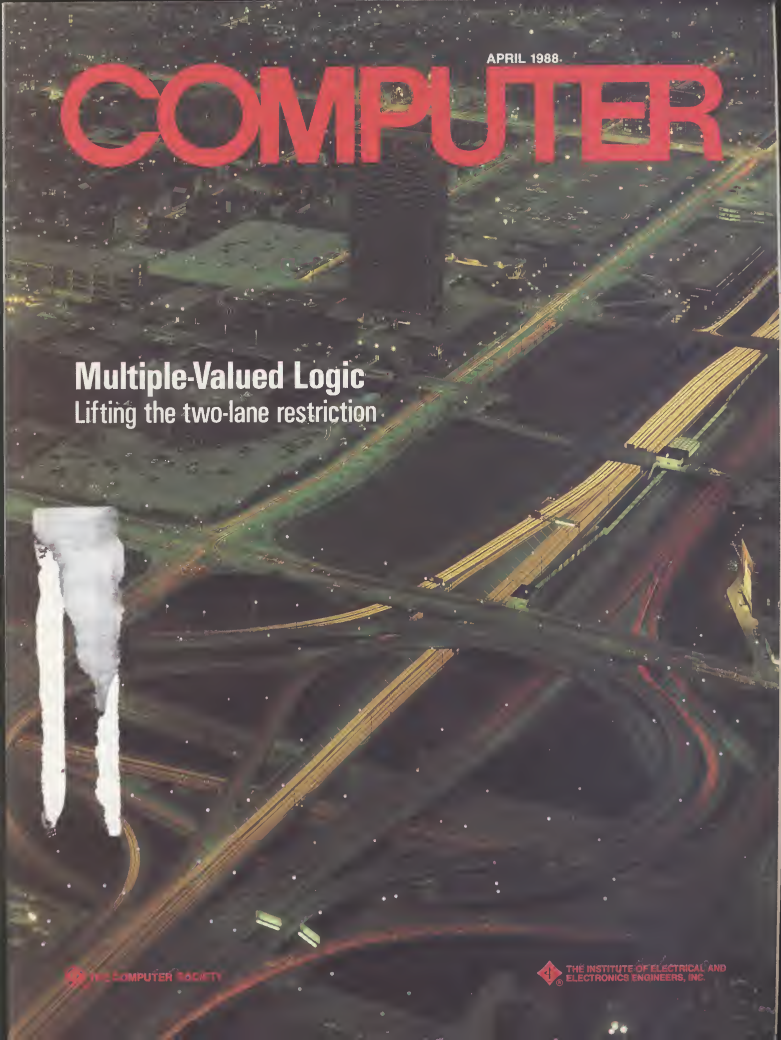
APRIL 1988

COMPUTER



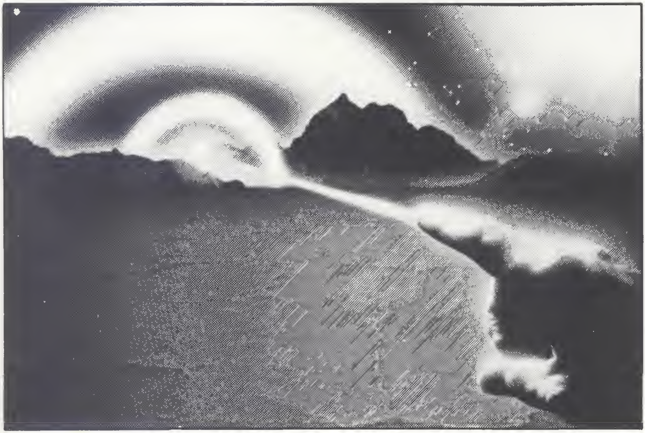
Multiple-Valued Logic

Lifting the two-lane restriction

THE COMPUTER SOCIETY

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Moving Information From One Era Into Another



At GTE's Computer and Intelligent Systems Laboratory, we've intensified our focus and created new opportunities in Software and Information Engineering and Artificial Intelligence. Here, we apply new ideas to research and development projects for improved information and telecommunication systems.

By expanding our scope and responsibility, we can better support GTE's telecommunications businesses. And our activities create challenges for individuals at the MS/PhD level in Computer Science. Join us as we continue to make history in telecommunications technology.

Areas of expertise:

- **SOFTWARE REUSABILITY**
- **PROGRAMMING ENVIRONMENTS**
- **SOFTWARE-DEFINED SERVICES FOR INTELLIGENT NETWORKS**
- **INFORMATION RETRIEVAL**
- **INTELLIGENT DATABASE MANAGEMENT SYSTEMS**
- **APPLIED EXPERT SYSTEMS**
- **EXECUTABLE SPECIFICATIONS**
- **RAPID PROTOTYPING**
- **ENTERPRISE ANALYSIS AND BUSINESS MODELING**
- **DISTRIBUTED ARTIFICIAL INTELLIGENCE**
- **LOGIC PROGRAMMING**
- **MACHINE LEARNING**
- **OPERATING SYSTEMS**
- **CONNECTIONIST MODELING**

GTE Laboratories offers attractive facilities located in a quiet, wooded setting just outside of Boston as well as a highly competitive salary and benefits package. We invite you to send a resume to Vanessa Stern, GTE Laboratories, Inc., Box IEEE4, 40 Sylvan Road, Waltham, MA 02254. An equal opportunity employer, M/F/H/V.

GTE Laboratories

EXCITING CAREER OPPORTUNITY

Design and develop a new simulation product

You must be a leader in a major application area that lends itself to simulation analysis

Robb Mills, developer of the telecommunication network analysis product--COMNET II.5

We need someone to develop the next addition to our family of simulation products--

- COMNET II.5-telecommunications network analysis
- NETWORK II.5-computer communications network design
- SIMFACTORY-factory planning and production analysis

Your knowledge-our technology

Your subject matter knowledge combined with our proven simulation technology and product experience, could be the beginning of an exciting new career for you.

You must have an advanced degree, understand discrete event simulation, and be a leader in a major application area.

Your career at CACI

As a Product Technical Manager you will be responsible for the new product.

You will define the product capabilities. Then you will develop the software, documentation, and training.

You will work closely with your colleagues in sales and marketing. This team will work together to make the product a commercial success.

Your compensation is high

Product Technical Managers are compensated with salary, a percentage of the product sales revenue, and a great working environment.

Our offices are near San Diego in La Jolla, California.

How to respond

Promptly call Joe Annino, President of CACI's Products Company, in complete confidence. Joe's number is (619) 457-9681.

Or send a resume and a letter describing your product idea, education, depth of expertise, and relevant experience to:

CACI
Products Company
3344 N. Torrey Pines Court
La Jolla, CA 92037
Attention: Joe Annino
(619) 457-9681

SIMSCRIPT II.5, NETWORK II.5, COMNET II.5, and SIMFACTORY are registered trademarks and service marks of CACI, INC.

©1988 CACI, INC.

COMPUTER

April 1988 **Published by the Computer Society** Vol. 21, No. 4

ARTICLES

13 **Multiple-Valued Logic: Guest Editor's Introduction and Bibliography**
Jon T. Butler

17 **Multiple-Valued Logic: A Tutorial and Appreciation**
Kenneth C. Smith
This tutorial places the developments and potential of multiple-valued signals and logic in the relevant context of binary and two-valued signals.

28 **Comparison of Binary and Multivalued ICs According to VLSI Criteria**
Daniel Etiemble and *Michel Israël*
How these ICs differ in terms of speed, chip areas, and power dissipation demonstrates their usefulness for differing applications.

43 **A Multiplier Chip with Multiple-Valued Bidirectional Current-Mode Logic Circuits**
Michitaka Kameyama, Shoji Kawahito, and *Tatsuo Higuchi*
We can achieve super chips using multiple-valued, bidirectional, current-mode logic circuits if we can solve the problems facing implementation.

58 **Multiple-Valued CCD Circuits**
Jon T. Butler and *Hans G. Kerkhoff*
Because CCD devices consume little power, we can realize complex circuits and increase their density with multiple-valued signals.

71 **Multiple-Valued Logic and Optimization of Programmable Logic Arrays**
Tsutomu Sasao
Multiple-valued input, two-valued output functions simplify the task of optimizing programmable logic arrays with input decoders.

83 **Fuzzy Logic**
Lofti A. Zadeh
Fuzzy logic—the logic underlying approximate, rather than exact, modes of reasoning—is finding applications that range from process control to medical diagnosis.

Circulation: *Computer* (ISSN 0018-9162) is published monthly by the Computer Society of the IEEE. IEEE Headquarters: 345 East 47th St., New York, NY 10017; Computer Society of the IEEE, Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 821-8380. Annual subscription included in society member dues. Nonmember prices: available upon request. Single copy prices: members $10.00; nonmembers $20.00. This magazine is also available in microfiche form.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854. Second class postage is paid at New York, New York, and at additional mailing offices.

Copyright and reprint permissions: Copyright © 1988 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons: (1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 29 Congress Street, Salem, MA 01970. (2) pre-1978 articles without fee. Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. For other copying, reprint, or republication permission, write to True Seaborn, Editor and Publisher, *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720. All rights reserved. Copyright © 1988 by the Institute of Electrical and Electronics Engineers, Inc.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computer* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space considerations. *Computer* subscribes to the Computer Press Association's code of professional ethics.

BPA

DEPARTMENTS

4 **President's Message**
Our international activities

6 **Letters**

99 **Standards**

102 **Computer Society News**
BoG and COPP reports; Compcon awards

106 **Update**
NSF budget

108 **New Product Reviews**
On buying a new 386-based PC

116 **New Products**

122 **IC Announcements**

123 **Microsystem Announcements**

126 **Conferences**
Compcon Spring 88

129 **Calendar**

135 **Call for Papers**

140 **Book Reviews**

143 **New Literature**

Traffic lights trace the multilevel, multipath interchange of three Chicago expressways—the Dan Ryan, Kennedy, and Eisenhower.

Cover photo: Fred Boler,
SuperStock International
Cover design: Jay Simpson,
Design & Direction

In the next issue

A spiral model of software development, modeling multicomputer systems, a remote computation facility, exploring algorithms, the optical file cabinet

Computer Society Information Page, inside back cover; Change-of-Address Form, 101; Membership Application Form, 115; Career Opportunities, 137; Advertising and Product Indexes, 144; Reader Service Card, 144A

COMPUTER
10662 Los Vaqueros Circle, Los Alamitos, CA 90720

Editor-in-Chief
Bruce D. Shriver, IBM T.J. Watson Research Center

Editorial Board
Dharma P. Agrawal, North Carolina State University
Scott Davidson, AT&T
Richard Eckhouse, MOCO
Gideon Frieder, Syracuse University
Edmund Gallizzi, Eckerd College
John Gurd, University of Manchester
James H. Haynes, University of California, Santa Cruz
Wiley McKinzie, Rochester Institute of Technology
K. M. Mohiuddin, IBM Almaden Research Laboratory
William Riddle, Software Productivity Consortium
Sallie Sheppard, Texas A&M University
Helen M. Wood, National Bureau of Standards
Akinori Yonezawa, Tokyo Institute of Technology

Submissions
Send six double-spaced copies of articles and special-issue proposals to Bruce D. Shriver, IBM T.J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598; (914) 789-7626. For electronic submission, Shriver's addresses are Compmail+, b.shriver; CSnet, Shriver@ibm.com; Vnet, shriver at yktvmh.

Staff
Editor and Publisher: True Seaborn
Assistant Publisher: Douglas L. Combs
Assistant to the Publisher: Pat Paulsen

Managing Editor: Marilyn Potes
Assistant Editors: Nancy Hays, Chuck Governale, Steve H. Wilcox

Art Director: Jay Simpson
Design/Production: Miriam Wiegel
Membership Manager: Christina Champion
Circulation Manager: Paul Zive
Advertising Director: Dawn Peck
Advertising Coordinators: Heidi Rex, Marian Tibayan

Magazine Advisory Committee
Sallie Sheppard (chair), Sumit DasGupta, Michael Evangelist, James J. Farrell III, Ted Lewis, David Pessel, True Seaborn, Bruce D. Shriver, John Staudhammer

Publications Board
James H. Aylor (chair), Victor Basili, J. Richard Burke, Jon T. Butler, J. T. Cain, Sumit DasGupta, Michael Evangelist, James J. Farrell III, Glen G. Langdon, Ted Lewis, Ming T. Liu, Theo Pavlidis, David Pessel, Sallie Sheppard, Bruce D. Shriver, Charles B. Silio, John Staudhammer, Harold Stone, Steven L. Tanimoto

Edward A. Parrish, Jr.

Computer Society President's MESSAGE

Strengthening our international activities

This month, the Computer Society takes another step forward in strengthening its role as an international organization.

The Executive Committee's third meeting of the year will be held April 11 in Brussels in conjunction with CompEuro 88. First held in Hamburg during 1987, CompEuro is cosponsored by IEEE's Region 8 and is one of many international conferences in which the Computer Society is involved.

Given this background, I would like to report to you the status of our international activities.

Membership and affiliate societies. Perhaps a good starting point is to reflect on the year-end 1987 membership statistics, as shown in Table 1. Regions 1-6 represent the United States, while Region 7 is Canada, Region 8 is Europe, Region 9 is South America, and Region 10 is Asia. Some 22 percent of our membership is in Regions 7-10. As this segment continues to grow, so should our opportunities for enhancing services available outside the continental United States.

As can be seen from Table 1, a portion of our total membership is made up of "affiliate" members. They are members of other professional organizations with which the society has established a formal affiliate relationship. At the present time, we are affiliated with 23 organizations (see box at right). About one-third of these organizations are in countries other than the United States, and that share may increase after the Brussels reception, where invitations to establish affiliate relationships will be extended to the European National Computer Societies. The Intersociety Coordinating Committee, chaired by Jon Butler of the Naval Postgraduate School, is charged with developing such relationships.

European and Asian offices. Recognizing the increasing importance of our non-US membership and the society's responsibility to provide services to

Table 1. Computer Society membership by regions, 1987.

Region	Students	Full members	Misc.*	Affiliates	Arrears	Total active	Percent change, active
1 (Northeastern US)	1,632	11,327	236	2,817	2,974	16,012	−0.4
2 (Mideastern US)	1,209	7,536	160	2,324	2,006	11,229	1.2
3 (Southeastern US)	1,230	4,879	125	1,113	1,688	7,347	0.9
4 (Midwestern US)	1,482	5,139	109	1,300	1,635	8,030	−2.3
5 (Southwestern US)	1,109	5,234	84	1,410	1,677	7,837	−0.7
6 (Pacific US)	1,774	13,231	331	3,898	3,488	19,234	1.2
Total US	8,436	47,346	1,045	12,862	13,468	69,689	0.2
7 (Canada)	799	3,079	72	484	1,116	4,434	−5.3
8 (Europe)	1,081	5,357	148	1,452	1,262	8,038	6.0
9 (South America)	437	854	85	133	648	1,509	−9.3
10 (Asia)	1,043	4,554	392	564	1,517	6,553	1.1
Total Non-US	3,360	13,844	697	2,633	4,543	20,534	0.6
Society total	11,796	61,190	1,742	15,495	18,011	90,223	0.3

*Full members who are retired, unemployed, or earn minimum wage.

those members, in January 1985 the Executive Committee adopted the goal of establishing a European office. By November of that year the office was in operation, in Brussels, with one employee. The Brussels office now has three employees and has moved to larger quarters. This office processes membership and subscription applications and distributes CS Press books with faster, better service than would be possible through the US offices. By accepting several European currencies, the office makes CS services much more easily accessible to members of the region. The staff of the Brussels office also assists in the organization of our European conferences and represents the society at conferences sponsored by other organizations.

Building on the success of the European operation, the Board of Governors recently authorized an Asian office in Tokyo. The executive director is currently working on implementation, with a goal of starting operation by mid-1988. The office will initially employ two staff and will offer services based on the Brussels model.

Distinguished Visitors Program. One of the successful activities that the society hopes to expand into the international arena is the Distinguished Visitors Program associated with chapters. This program brings renowned speakers to chapter meetings at no cost to the chapter.

As shown in Table 2, there are already some 63 chapters in Regions 7-10, which should offer a base for launching such a program. In fact, some activity of this nature has been underway for some time.

Region 10 (Asia) had a successful program from about 1981 to 1985, with an average of three or four speakers each year who agreed to make a lecture tour through Asia. The society has appropriated funds in the 1988 budget to reactivate this program.

Conferences. Another major international activity involves conferences sponsored or cosponsored by the society. While they are far too numerous to list, let me cite a few examples to give you an idea of the spectrum of topics and cooperating organizations:

- International Conference on Computer-Aided Design — cosponsored with the IEEE Circuits and Systems Society
- Fault-Tolerant Computing Symposium — cosponsored with the Institute of Electronics, Information, and Communication Engineers of Japan
- International Conference on Computer Applications — cosponsored with the Chinese Computer Society
- International Conference on Distributed Computing Systems
- International Conference on Software Engineering — cosponsored with ACM and the National Computer Board of Singapore

By sharing this information, I hope to have given you a new perspective on your society. In a profession such as ours, where rapid technological breakthroughs have become nearly commonplace, it is important to provide as many communication paths as possible among its practitioners. Thus, the additional conference and chapter activities, as well as the new society offices, are all aimed at providing these paths to all of our members.

Edward A. Parrish, Jr.
Computer Society President

Table 2. International chapters.

Region	7	8	9	10
Regular chapters	11	7	6	11
Student chapters	10	2	4	12

Approved affiliate societies

American Institute of Aeronautics and Astronautics (AIAA)
American Institute of Physics (AIP)
American Mathematical Society (AMS)
American Society of Mechanical Engineers (ASME)
Association for Computing Machinery (ACM)
Australian Computer Society
British Computer Society
Computer Engineering and Application Society of the Chinese Institute of Electronics (CEAS)
Computer Society of the Republic of China (Taiwan)
Data Processing Management Association (DPMA)
Information Processing Society of Japan (IPSJ)
Institute of Electrical Engineers of Japan (IEE-JAPAN)
Institute of Electronic, Information, and Communication Engineers (IEICE) of Japan
Institution of Electrical Engineers (IEE-UK)
Instrument Society of America (ISA)
Mathematical Association of America
National Association of Accountants
Operations Research Society of America (ORSA)
Society of Aircraft Materials and Processes (SAMP)
Society of Automotive Engineers (SAE)
Society for Computer Simulation (SCS)
Society for Industrial and Applied Mathematics (SIAM)
Society for Photo-optical Instrumentation Engineers (SPIE)

LETTERS TO THE EDITOR

Does method invalidate results?

To the editor:

There are serious problems with the special report, "The Effect of Abstract Data Types on Program Development," by Mitchell, Urban, and McDonald in the August issue of *Computer*. In fact, the results of the authors' experiments demonstrate absolutely nothing with respect to the efficacy of abstract data types as a program development tool.

To summarize the experiment that was conducted: The test subjects received some initial training in design and coding of programs. They were then divided into control and experiment groups of only four subjects each, all of whom are male. The experiment group received additional training of only one hour in another design methodology. The two groups were then given a single test task, and it was found that the performance of the experimental group had deteriorated relative to both the control group and their own past performance.

First of all, the data gathered is biased by a number of factors: All of the subjects are male. They are all volunteers. They have had a small number of previous instructors. They are all students at one university, and they have all been taught under the same curriculum. These factors are a direct result of the small sample size and the narrowness of the population cross-section that was chosen. The authors show that their results are statistically significant, but they must also demonstrate that their sample is representative of the general population to draw any general conclusions. For example, the subjects had an average of five previous computer science courses but had not yet been exposed to abstract data types as a programming methodology. This contrasts sharply with the curriculum of schools that teach data abstraction techniques beginning with the first or second course. One could thus argue that the authors' results are not representative of the latter portion of the population. That same argument can be extended to reach the conclusion that only a very small portion of the general population is actually represented by the authors' data.

One can also argue that a single programming assignment is not generally representative of abstract-data-type implementation tasks. The authors fail to specify the details of the test assignment; however, it can be assumed from the short amount of time required for the subjects to complete it that the assignment was not particularly complex. Performance is often dependent on the complexity of the task, and thus it is necessary to administer tests at various levels of complexity to determine whether such a dependency is present. This is especially true when testing the effect of an organizational method, because such methods often incur an overhead factor that reduces performance on simple tasks, despite improved performance on complex tasks.

In addition, the training received by the experimental group is questionable from a pedagogical viewpoint. Unfortunately, the authors only vaguely specify the content of the training. However, one hour is hardly a sufficient amount of time in which to adequately teach data abstraction techniques. Many curricula specify an entire course for the subject, with additional training and practice in subsequent courses. Consider that several hours may be required simply to teach the syntax of the Ada package facility, which is intimately related to the implementation of abstract types in that language.

Most important, however, is that this experiment follows precisely the same scenario as the classic psychological experiments that demonstrated the principle of proactive inhibition. That is, pretraining on one task inhibits subsequent learning of a different but related task. Also, by administering just one post-training test task, the authors only sampled the subjects performance at the lowest point in the learning curve. For the results to be valid, it would have been necessary for the authors to administer additional training tasks until the performance of the experimental group stabilized.

An analogous situation is seen in the retraining of a typist to use an ergonomically arranged keyboard. The typist may initially suffer a 50 to 75 percent decrease in performance, followed by an eventual net increase in performance of as much as 100 percent. The method used by the authors only detects the initial decrease and does not determine the ultimate relative performance of the experimental subjects after they have become proficient in applying the new technique.

Thus, the authors' conclusion that manual implementation of abstract data types generally results in reduced programmer productivity and program efficiency is not at all supported by their experimental data. The data merely serve to confirm the well-known fact that psychological models of learning apply as much to programming as they do to other learned skills.

In summary, when testing human performance, it is essential that a representative sample of the population be chosen. When the testing is applied to a newly learned skill, it is equally important that the psychological effects associated with learning be taken into consideration, and that the testing method be designed accordingly. The effects of different pedagogical approaches to the training must also be accounted for if the intent is to draw general conclusions. Finally, it is important to keep in mind that the enormous variability of humans usually makes it difficult to draw such simple, unquali-

Computer welcomes your letters.
Send technical correspondence to Bruce D. Shriver, *Computer* Editor-in-Chief, IBM T.J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598. Send other comments to Letters Editor, *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720.
All submissions are subject to editing for style, length, and clarity.

CAPTURE YOUR WILDEST IMAGININGS

Come to SIGGRAPH '88 in Atlanta where computer graphics trailblazers will take you on a great adventure.

Explore the diverse ways graphics are used in science, academia, industry and the arts. Listen to computer graphics pioneers describe bold, new advances. Learn how to give life to your most daring ideas.

Venture into the era of electronic visualization.

ACM SIGGRAPH '88
15th Annual Conference
on Computer Graphics
and Interactive Techniques
Atlanta, Georgia
August 1-5, 1988

Sponsored by the
Association for Computing
Machinery's Special
Interest Group on
Computer Graphics in
cooperation with the
IEEE Technical Committee
on Computer Graphics.

Please send a copy of the SIGGRAPH '88 advance program to:

Name

Address Company

City State/Country

Zip Code Telephone

Members of ACM SIGGRAPH will automatically receive these materials in the mail in April and need NOT return this coupon.

ACM SIGGRAPH '88
Conference Management
111 East Wacker Drive, Suite 600
Chicago, IL 60601
(312) 644-6610

Image: RGB Computer Graphics Service–M. Ontani / B. Spadacini / Milan / Italy

fied conclusions about their performance on a complex test. We hope that in their future research, the authors will take these factors into consideration.

Chip Weems
University of Massachusetts at Amherst

Nell Dale
University of Texas at Austin

Author's reply:

Weems and Dale's thoughtful letter provides a nice summary of a few of the trade-offs and pitfalls empirical researchers must thoroughly evaluate in the design and execution of any experiment. To summarize, they are concerned first with the generalizability of our findings, including the number and demographics of the subjects, and second with the design of the experiment, especially the effect of the treatment.

As reported in "The Effect of Abstract Data Types on Program Development," participation required a serious time commitment of more than three months for each subject involved in the experiment. This cost, together with the required level of programming sophistication, made subject recruitment difficult. In fact, we did not select a sample of a population of professional programmers and hence did not generalize the results to that population.

Our article does provide an interesting perspective on abstract data types as well as some significant empirical evidence that the code required to support abstract data types should be prepared in advance or generated automatically. These findings apply to the group of programmers we tested for the specific treatment and instruments used. It would be both interesting and *costly* to conduct a similar experiment on a representative sample of some large population of professional programmers controlling for demographics and other extraneous factors.

We were very aware of each of the confounding factors described by Weems and Dale. To control such problems we employed a standard, widely used experimental design (the pretest, post-test control group design of Campbell and Stanley, *Experimental and Quasi-Experimental Designs for Research*, 1966). The design includes the random assignment of subjects to groups and controls for the effect of the treatment. In addition, various controls were in effect throughout the experiment to insure that the performance of the subjects was not influenced by the investigators and to isolate the effect of the treatment.

Weems and Dale have arrived at several unsupported conclusions regarding the background of the subjects, the treatment, and the effect of the treatment. They ascribe the deterioration of the control group's performance to unfamiliarity with the new technique. In fact, the subjects were taught to use simple aliasing strategy, which allowed them to postpone the implementation of data types. The treatment session continued until *all* the subjects in the experimental group were proficient in the use of the new methodology. We are familiar with the learning curve and its possible influence, and believe that the results as reported reflect the effect of the treatment and were not due to unfamiliarity with the new methodology.

Finally, we do agree with Weems and Dale in realizing that empirical research is especially costly and that experimental research requires strict controls over extraneous factors so that the effect of the independent variables can be isolated. We also agree that generalizability can be a major difficulty and that, if the results of a study are to be generalized beyond the actual subjects, suitable sampling procedures must be employed to assure that the sample is representative of the population.

Jeffrey Mitchell
University of Southwestern Louisiana

Formal methods and source code—a conflict?

To the editor:

"Source Code" by Mark Weiser (November 1987 *Computer*, pp. 66-73) is full of misconceptions and inaccuracies, and I would like to discuss those about formal methods.

He writes (p. 67), "having code around means that you are closer to the truth about your system — closer than any manual or proof can bring you." I do not agree with this. A manual has a different purpose than a program. It should explain things to people, while the program's main purpose is to state instructions to a machine. If the user has a good manual he doesn't need to be "closer to the truth." With proofs we have a different situation because a proof does say more about your system.

There are several kinds of proofs, each being about different properties of a program, for example, termination. The most common case is correctness verification, sometimes separated as partial correctness and total correctness. Partial correctness is verified proving that the code satisfies a formal specification. This is done by mathematical calculation. The specification is not necessarily the requirements specification but may be derived from it. At this point the proof contains at least three things: a specification, the program, and the verification. The specification tells the user about properties of the result produced by the program; this is *what* the program does. The program tells the computer *how* this is done. The verification proves that the program does indeed *satisfy* some given properties. All of this cannot be said from the code alone.

Formal verification as just described, an after-the-fact calculation, is only one way to carry it out. A very promising (and pleasant) technique is to develop the code along with the proof. This is better known as formal derivation (see C.A.R. Hoare's article, "An Overview of Some Formal Methods for Program Design," September 1987 *Computer*, pp. 85-91).

Weiser continues and criticizes formal proofs (p. 68):

> ... the abstract interface is a formal description of the program, and the interface has been 'proven' to be a correct description of the program. Unfortunately, program proofs are unreliable descriptions for two reasons. First, ... because they do not have the cultural checks and balances accorded proofs in mathematics, and even proofs in mathematics are unreliable. ... Second, ... mathematics is not the correct kind of description for programs. It describes a different world, containing different objects, in a different language, than is executed by machines.

Again, I disagree. You don't prove that a specification is a correct description of a program; it is the other way around. You prove that the program is a correct implementation of its specification. And, if what you want is to prove that a specification describes the program's requirements, you can't do it. Quoting from Hoare's article (p. 85):

> ... ensure that the formalization of requirements is simple and well-structured so that, together with its informal explanation, it obviously describes what is really wanted, and not something else. There can never be any formal method of checking this vital correspondence, because there should not be any more obviously correct description to check it against.

The reasons Weiser gives to show that mathematical proofs are unreliable are flawed. His "checks and balances" and

COMPUTER SCIENCE

from Academic Press Journals

JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING

Editors-in-Chief: **Kai Hwang**
University of Southern California, Los Angeles

Leonard Uhr
University of Wisconsin, Madison

This international journal is directed to researchers, engineers, educators, managers, programmers, and users of computers who have particular interests in parallel processing and/or distributed computing. The *Journal of Parallel and Distributed Computing* publishes original research papers and timely review articles on the theory, design, evaluation, and practices of parallel and/or distributed computing systems.

Volume 5 (1988), 6 issues ISSN 0743-7315 In the U.S.A. and Canada: $123.00 All other countries: $160.00

Reader Service Number 1

Also from Academic Press

JOURNAL OF COMPLEXITY

Editor
Joseph F. Traub
Columbia University, New York City

Volume 4 (1988), 4 issues
ISSN 0885-064X

In the U.S.A. and Canada: $84.00
All other countries: $101.50

INFORMATION AND COMPUTATION

Editor
Albert R. Meyer
Massachusetts Institute of Technology, Cambridge

Volumes 76–79 (1988), 12 issues
ISSN 0890-5401

In the U.S.A. and Canada: $384.00
All other countries: $468.00

JOURNAL OF COMPUTER AND SYSTEM SCIENCES

Managing Editor
E. K. Blum
University of Southern California, Los Angeles

Volumes 36–37 (1988), 6 issues
ISSN 0022-0000

In the U.S.A. and Canada: $288.00
All other countries: $333.00

For more information, please write or call:

ACADEMIC PRESS, INC.
Journal Promotion Department
1250 Sixth Avenue
San Diego, CA 92101 U.S.A.
(619) 230-1840

Please enter my subscription for:

☐ Journal of Parallel and Distributed Computing, Vol. 5 (1988)
☐ Journal of Complexity, Vol. 4 (1988)
☐ Information and Computation, Vols. 76–79 (1988)
☐ Journal of Computer and System Sciences, Vols. 36–37 (1988)

Please send me:

☐ A sample copy of ____________
☐ Privileged personal rate information. Personal subscription rates are available only on orders placed directly with the Publisher and paid for with personal funds. Checks and money orders must be drawn against a U.S. bank.

Method of payment:

☐ Our purchase order is enclosed.
☐ My check is enclosed.
☐ Credit information is given.
☐ Enter as a Standing Order (institutions only) hereby authorizing Academic Press to service and bill our account each calendar year until cancelled.

Please note: Payment or credit card information must accompany order. Prices include postage, handling, and air freight where applicable, and are subject to change without notice. Payment must be in U.S. currency, U.S. Bank Draft, International Money Order, or UNESCO coupons.

When paying by credit card, please check one box:

☐ Visa/Barclaycard
☐ MasterCard/Access
☐ American Express
☐ Diner's Club

Expiration Date (Mo/Yr): ____________

Credit card number: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Signature: ____________

Total Amount: $ ____________

Name: ____________

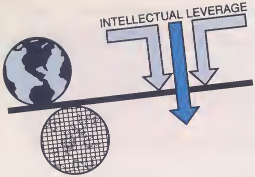
Address: ____________

____

City: ____________ State: ________ Zip: ________

Country: ____________

S8087



Your call for participation

in COMPCON SPRING 89

The year's only broad-based computing conference
by the world's largest computer society.

Start Your Participation in COMPCON SPRING 89 Now!

WHAT IS COMPCON? Over the years COMPCON has been known for consistently providing a high quality program covering broad-based updates on technological progress. Each year, technical people with transnational contacts from industry, academia and research labs get together and study the rapid advances being made in the computer field.

EXPAND YOUR PROFESSIONAL GROWTH! Here is your opportunity to contribute toward making this major computing conference the best COMPCON ever! Assist the Program Committee in generating a program which includes topics in your area of expertise or interest.

1. Submit an extended abstract, or a full paper, in your area of expertise for a proposed COMPCON presentation. Send four copies and include your phone number.
2. Submit a proposal for a session, panel or tutorial you would like to organize or in which you would like to participate.
3. Suggest topics, sessions, panels or tutorials you would like to have the Program Committee develop and present, if possible, at COMPCON SPRING 89.

To maintain high quality, the Program Committee reviews all papers and generally limits sessions to ninety minutes with three speakers. Papers on your experiences with real systems rather than those with theoretical contencts are strongly encouraged.

CONTACT:

Dr. Kenichi Miura (COMPCON 89 Program Chair)
Computational Research Department
Mail Stop B2-7
FUJITSU AMERICA, INC.
3055 Orchard Drive
San Jose, CA 95134-2017
(409) 432-1300 ext 5408 or 5723

WRITE NOW!
The earliest submissions have the most influence.
The submission deadline is June 1, 1988.
Notificiation of acceptance will be in October, 1988.

Hot Topic Areas

- *Highly Parallel Systems*
  - Parallel Architectures
  - Super Computers and Minisupers
  - Parallel Algorithms
  - Fault Tolerant Systems
  - Data Flow Machines
  - Advanced Scientific Computing
  - The New Workstation
- *Distributed Systems and Networks*
  - Distributed Operating Systems
  - Local Area Networks
  - Wide Area Networks
  - Distributed File Systems
  - Factory and Office Automation
  - Distributed Data Bases
  - Data and Voice Digital Networks
  - New Network Protocols
- *Knowledge Engineering*
  - Expert Systems
  - Logic Programming and Inference Engines
  - AI Architectures
  - Robotics
  - Computer Vision
  - Data Bases
- *Software Engineering*
  - Managing Software Complexity
  - Unix
  - Graphics
  - Quality Software: How to get it
  - User Interfaces
  - CAD Tools
  - Object Oriented Languages, Data Bases and Systems
  - Window Systems
  - Data Security and Encryption
- *Emerging Technologies*
  - The 5th Generation Computers
  - Optical Computing
  - Neural Based Architectures
  - New ~~Architectural Con~~troversies
  - New Microelectronics Technologies
- *If we missed your topic of interest, please suggest it*

February 27 - March 3, 1989

THIRTY-FOURTH IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE

Cathedral Hill Hotel, San Francisco, California

February 27 – March 3, 1989 spring compcon 89

San Francisco

SPONSORED BY THE
IEEE COMPUTER SOCIETY
THE INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS, INC.

his statement (p. 69) that "A proof must stand the test of time" do not say anything about mathematical proofs; they apply only to the people that make them. A proof is either correct or incorrect; time has nothing to do with it. Then, his second reason is further supported by the phrase (p. 69), "A computer program is not a formal object, nor is computation formal." Again, quoting Hoare (p. 85), "The code of a computer program is a formal text, describing precisely the actions of a computer executing that program." In fact, as you can also see in Hoare's article, the computer program is written in one of the most restricted of formal notations. Another is Boolean algebra, routinely used to design computers. ...

In conclusion, the article has many misconceptions — and not only about formalism. ... As an engineer I have always liked having a sound theoretical foundation behind my work. ... Just knowing that the procedures and practices of this profession have a solid background is reassuring, even if it is not complete or perfect.

René Berber
Mexico City

Author's reply:

René Berber makes some excellent points with which I agree. Yes, mathematical theories of programs are interesting and useful. Yes, establishing a relation between the two formal objects, a specification and a program, is often very revealing. Yes, having manuals and formal descriptions of a program, in addition to the source, are better than source alone. I presume that Berber would agree with me that having the source to a program is also important — if for nothing else, so there is something to prove correct.

Berber, quoting Tony Hoare, would say, "The code of a computer program is a formal text." Yes, the code of a computer program can be viewed as a formal text. But it can also be viewed as a part of the control system for a running computer. The two views have a relationship, with the formal view of secondary importance. The formal correctness of the code is scant comfort if my airplane crashes because the computer exercises a bug in the CPU or it is struck by lightning.

Programs are not fundamentally, or even most interestingly, formal texts. If that is all they were, their study would a branch of pure mathematics. In my view, programs are most fundamentally "things" of the world, and the attention given their formal properties is mostly because of their ability to have a physical presence in our surroundings. The most important conclusion of my article is that specifications, formal or otherwise, of programs cannot replace access to the programs themselves. Formal properties are important, as Hoare says earlier in the same article, because of the hope "that conclusions drawn from analysis...will apply...to software engineering projects on a more realistic scale." It is the real software projects that are the master, the formal analysis the servant.

It is a wonderful and useful property of programs that they can be viewed as formal texts. But a program is much more than its formally verifiable properties. Like the person looking for his wallet under the streetlamp because the light is better there, we may never find what we seek of our programs if we look only at what can be formally specified.

Mark Weiser
Xerox Palo Alto Research Center

SOFTWARE DEVELOPERS

Have an Impact on the Rapidly Growing Electronic Publishing Industry

Linotype Company, formerly known as Mergenthaler, and recognized throughout the world as a prime innovator in the design and development of computerized typesetting and photocomposition systems, has outstanding career opportunities available for individuals with various degrees of experience in the computer and graphic arts field.

PROJECT LEADER
Requires a Bachelor's degree in C.S., E.E. or equivalent education/work experience. A minimum of 10 years experience in software development for complex embedded systems. Candidates will be expected to assist in development of prespecs, external specifications and overall design. Will manage a project staff of 5 people, providing both technical and management guidance to the project team. For a moderately complex project, will define the project development in terms of tasks, milestones, budget, and resources required. Will track and maintain project development and provide timely communication to upper management on project status. Knowledge of C and UNIX required, in addition to experience in development of user interface software and text editors. Master's degree in C.S. and experience in graphic arts industry a plus.

SOFTWARE ENGINEERS
Requires a Bachelor's degree in C.S., E.E. or equivalent education/work experience. Requires a minimum of 5 years experience in software development. Candidates will be expected to design complex software systems from external specifications and to conduct project design reviews. Will assist the senior staff in preparation of external specifications as well as providing technical guidance to the lower level staff. Assist the project leader in coordination of product development. Knowledge of C and UNIX required. Experience in development of user interface software and text editors is necessary.

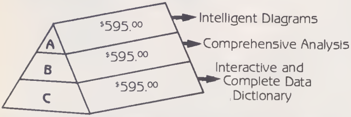
We offer a professional growth environment, as well as excellent salaries and benefit package. For immediate consideration, please send your resume, including salary history/requirements, in confidence to:

Director, Human Resources
425 Oser Avenue
Hauppauge, New York 11788
Equal Opportunity Employer

THE MODULAR IBM PC BASED

VISIBLE ANALYST® WORKBENCH

THE ABC'S OF CASE



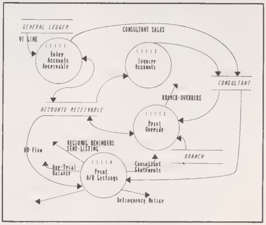
STEP 1 **The Visible Analyst® - Module A of The Workbench.**
Intelligent free form or grid locked graphic/text diagramming, using 21 pre-programmed DFD and flowchart symbols, with the ability to generate and use additional symbols. User created symbols can be made out of any combination of lines, arcs, boxes, circles and shading. In addition each symbol can be stylized, i.e., overlay, sized, reliefed and bolded, multiple line styles and text fonts, page size of 15" x 11" or 8½" x 11" with unlimited nested decompositions. Proprietary pixel mapper for high quality presentation output (180 dpi) and much more . . .

STEP 2 **The Visible Rules - Module B of The Workbench.**
Expert Rules analyzes a DFD or set of DFD's produced with the Visible Analyst. Generates error messages if it does not conform to the syntax of the Yourdon/DeMarco or Gane/Sarson techniques. Features include data flow splitting, automatic level balancing and automatic population of the data dictionary. Export/Import in ASC II format for easy linking to host system.

STEP 3 **The Visible Dictionary - Module C of The Workbench.**
Automatic creation of entries from diagrams, automatic location referencing, wild card search and finds, specialized dictionary screens for different symbols, multiple system reports, tightly coupled and interactive to Module A and B.

STEP 4 **Multi-User Workbench - Coming Soon**
The power of structured methodologies, user-friendly graphics, comprehensive analysis, and common data dictionary will combine under a Novell Netware LAN. Customers on our update and maintenance program are eligible for a free upgrade to the LAN System.

STEP 5 **Update/Maintenance Program**
$50. per Tool per Year. Stay tuned with the EVOLUTION IN SOFTWARE AUTOMATION with our low cost Update/Maintenance program.



YOURDON DIAGRAM

"Visible Analyst Workbench is a most impressive product"

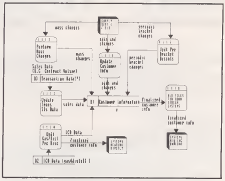
Computer Language Magazine

"Visible Analyst Workbench is a useful tool at a reasonable price"

PC Week

"Visible Analysts' diagramming capabilities are specifically tailored to the needs of system developers"

IEEE Computer Magazine



GANE & SARSON DIAGRAM

SEE US

Come to one of our free "hands-on" workshops around the country. Our product will be installed and available for use!!! Call to register.

TRY US

Take our product for a 30 day trial. Cost is $200.00. Fully functional product, tech support, and full documentation included. $200 is applicable to purchase price.

BUY US

$595.00 PER MODULE

And join the hundreds of satisfied customers already slashing development costs.

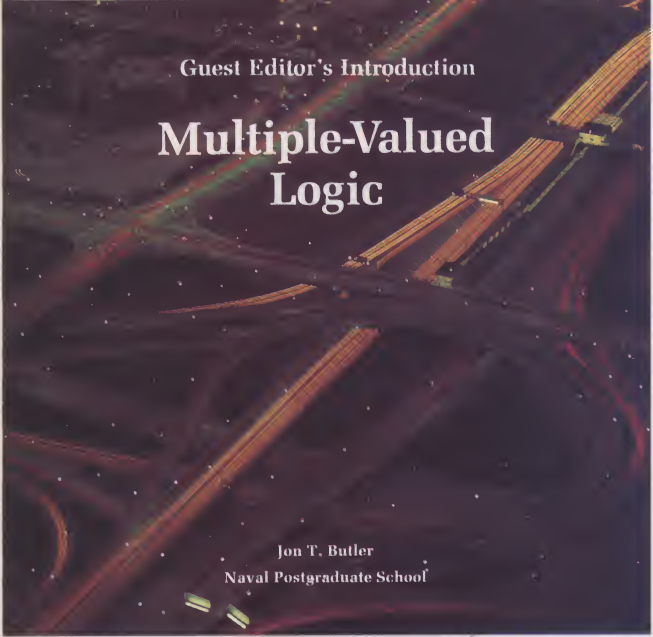
START EXPERIENCING *THE VISIBLE SOLUTION*™ TODAY!!!

VISIBLE SYSTEMS CORPORATION

Phone: (617) 969-4100 Telex: 261102 VS UR
49 Lexington Street Newton, MA 02165

© 1988 Visible systems Corporation Visible, The Visible Analyst and Visible Systems Corporation are registered trademarks of Visible Systems Corporation

Reader Service Number 2



Guest Editor's Introduction

Multiple-Valued Logic

Jon T. Butler
Naval Postgraduate School

One of the topics in a first course on computing is the binary number system. Among a student's first realizations is that a large number of bits is needed to represent even moderately large numbers. For example, the decimal number 936 is represented in binary by ten symbols (1110101000). Also, when adding more than two binary numbers, carries often extend beyond the next bit position. For example, the sum of 1+1+1+1, which is 100 in binary, involves a carry out of the least significant bit into the second bit over. It is easy to see why our ancestors preferred decimal. No intelligent computerless civilization would ever adopt binary.

In recent years, researchers have questioned the use of binary by an intelligent computerized civilization. With binary, there is inefficient use of the interconnect, a large part of any computing system. For example, about 70 percent of the area in a very-large-scale-integration (VLSI) chip is devoted to connections among devices. A significant savings in chip area can be realized through the use of more than two levels of logic. For example, a four-valued signal on a single lead carries twice the information carried by a binary signal. Also, multiple-valued logic is useful in reducing the wiring between chips and stands as a solution to the pinout problem, where an insufficient number of pins restricts the flow of information into and out of the chip. Restricting a system to binary is like restricting automobile traffic to two-lane highways.

Multiple-valued signals result in more compact circuits in other ways, as well. For example, the use of logic levels that correlate directly to gray levels in image processing systems eliminates the need for decoder circuits. Also, a base 10 logic system requires no decoders for decimal input and output data.

Besides its direct use in ICs, multiple-valued logic has had indirect use, as well. Multiple-valued logic is used in the analysis and design of binary systems. For example, it is used in the minimization of binary programmable logic arrays. Also, multiple-valued logic concepts have been used to locate faults in multiprocessing systems.

There is a growing exploitation of the advantages of multiple-valed logic in commercially available ICs. In 1980, multiple-valued circuits appeared in Intel's 8087 numeric coprocessor and in the iAPX-432 microprocessor. Each contained a four-valued read-only memory. These were followed by the use of four-valued ROMs in Motorola and General Instrument semicustom ICs. Prototype chips include an arithmetic multiplier, an image processing

0018-9162/88/0400-0013$01.00 © 1988 IEEE

array, and a programmable logic array. Fuzzy logic chips have been built which process a logic that has essentially infinitely many values. These include a fuzzy inference chip by Bell Laboratories and a fuzzy logic computer in Japan.

Our goal in this special issue is to acquaint you with recent developments in multiple-valued logic. The main theme is VLSI implementation. There are other important aspects, including algebra, fault-tolerant computing, and optical computing. The annotated bibliography accompanying this introduction is a pointer to these other areas.

This special issue serves as a benchmark for successes and expectations in 1988. Future contributions will depend on the energies of researchers in the area and on acceptance by researchers outside the area. Today's wide use of binary in computer systems follows its wide use in the past. For historical reasons, we continue to adopt binary, passing it to future generations through the curricula of our schools and universities. However, binary logic, like von Neumann architecture or Fortran programming, is widely used and well-entrenched, but replaceable by something better. ☐

Jon T. Butler is a professor of electrical and computer engineering at the Naval Postgraduate School in Monterey, Calif. His research interests include multiple-valued logic, computer-aided design, and reliability of multiprocessing systems.

Butler was chair of the Computer Society's Multiple-Valued Technical Committee from 1980 to 1981. He has served as an editor of the *IEEE Transactions on Computers* and as an editor for the Computer Society Press. He has been a distinguished visitor of the Computer Society, and has served as the vice chair for Hardware of the Technical Activities Board. He has been the NAVELEX Chair Professor of the Naval Postgraduate School and has held two separate National Research Council Postdoctoral Associateships.

Butler received a PhD degree from Ohio State University in 1973. He received ME and BEE degrees from Rensselaer Polytechnic Institute in 1967 and 1966, respectively.

Readers may write to Butler (Code 62-Bu) at the Dept. of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA 93943-5100.

Further reading

Readers wanting more information on multiple-valued logic can consult the following survey/tutorial articles. Also listed are special issues covering both narrow and wide ranges of topics within multiple-valued logic. This bibliography includes topics and associated papers not covered by articles in this special issue. An asterisk (*) denotes an easily read item.

Survey and tutorial papers/texts

A good introduction for the beginner:

- * Muzio, J.C., and T.C. Wesselkamper, *Multiple-Valued Switching Theory*, Adam Hilger, Bristol and Boston, 1986.

Two easily read survey/tutorial papers:

- * Hurst, S.L., "Multiple-Valued Logic—Its Status and Its Future," *IEEE Trans. Computers*, Dec. 1984, pp. 1160-1179.
- * Smith, K.C., "The Prospect for Multivalued Logic: A Technology and Applications View," *IEEE Trans. Computers*, Dec. 1981, pp. 619-632.

A survey and a description of contributions to multiple-valued logic by country, by new authors, etc.:

- * Ginzer, J.A., and J.T. Butler, "Multiple-Valued Logic: 1974-1978 Survey and Analysis," *Proc. 9th Int'l Symp. Multiple-Valued Logic*, May 1979, pp. 1-13.

Multiple-valued logic has been useful in realizing compact memories, a topic of the following paper:

- * Rich, D.A., "A Survey of Multivalued Memories," *IEEE Trans. Computers*, Feb. 1986, pp. 99-106.

A collection of important papers on various disciplines within multiple-valued logic:

Rine, D.C., ed., *Computer Science and Multiple-Valued Logic: Theory and Applications*, second edition, North-Holland, 1984.

Special Issues devoted to multiple-valued logic

The following issues are listed in order of publication, the most recent first:

Hawkin, R.E., ed., *Int'l J. of Electronics*, Aug. 1987.

Muzio, J.C., and I.C. Rosenberg, eds., *IEEE Trans. Computers*, Feb. 1986.

Arrathoon, R., ed., *Optical Computing*, Jan. 1986.

Butler, J.T., and A.S. Wojcik, eds., *IEEE Trans. Computers*, Sept. 1981.

Vranesic, Z.G., ed., *IEEE Trans. Computers*, Dec. 1977.

- * Rine, D.C., ed., *Computer*, Sept. 1974.

Algebra

Algebra is the keystone of modern multiple-valued logic. It forms the basis on which decisions are made as to which functions to provide the user of a multiple-valued logic. The following is a good survey of algebra, with over 400 citations (it is difficult to read):

Rosenberg, I.G., "Completeness Properties of Multiple-Valued Logic Algebras," in *Computer Science and Multiple Valued Logic: Theory and Applications*, second edition, D.C. Rine, ed., North-Holland, 1984, pp. 144-186.

An archival paper in this area:

Epstein, G., "The Lattice Theory of Post Algebras," *Computer Science and Multiple-Valued Logic: Theory and Applications*, second edition, D.C. Rine, ed., North-Holland, 1984, pp. 23-40.

Optoelectronics

A sister technology for which multiple-valued logic has application is optoelectronics. The promise of both is efficient computation through improved interconnect. The following three papers represent, respectively, a summary, a survey, and a proposal for multiple-valued optical systems.

* Arrathoon, R., "Optical Multiple-Valued and Multiple-Class Discrimination," *Proc. 16th Int'l Symp. Multiple-Valued Logic*, May 1986, pp. 146-147.

* Hurst, S.L., "A Survey: Developments in Optoelectronics and its Applicability to Multiple-Valued Logic," *Proc. 16th Int'l Symp. Multiple-Valued Logic*, May 1986, pp. 179-187.

Mirsalehi, M.M., and T.K. Gaylord, "Content-Addressable Memory Processing: Multilevel Coding, Logic Minimization and an Optical Implementation," *Proc. 16th Int'l Symp. Multiple-Valued Logic*, May 1986, pp. 174-178.

Spectral logic

In spectral logic, functions are represented as a set of numbers. This gives a better understanding of the properties of functions and insight into better ways to realize them. The following gives a useful application:

Moraga, C., and K. Seseke, "The Chrestenson Transform In Pattern Analysis," in *Spectral Techniques and Fault Detection*, M. Karpovsky, ed., Academic Press, Orlando, Fla., 1985, pp. 143-178.

Archival papers:

Chrestenson, H.E., "A Class of Generalized Walsh Functions," *Pacific J. Math.*, 1955, pp. 17-31.

Karpovsky, M., "Spectral Methods for the Decomposition, Design, and Testing of Multiple-Valued Logic Networks," *Proc. 11th Int'l Symp. Multiple-Valued Logic*, May 1981, pp. 1-9.

Testing

As in binary, there is a strong interest in reliable circuits. The next two papers are archival papers in the area of testing of multiple-valued circuits:

Druzeta, A., and Z.G. Vranesic, "A Higher Radix Technique for Fault Detection in Many-Valued Multithreshold Networks," *IEEE Trans. Computers*, Nov. 1978, pp. 1070-1073.

Etiemble, D., "Multivalued I^2L Circuits for TSC Checkers," *IEEE Trans. Computers*, pp. 537-540, June 1980.

The modeling of faults in a binary circuit can be facilitated by a multiple-valued system, which is the topic of the following paper:

Hayes, J.P., "Digital Simulation with Multiple Logic Values," *IEEE Trans. CAD*, Apr. 1986, pp. 274-283.

Systems diagnosis

Multiple-valued concepts are useful in analyzing multiprocessor systems where processors produce tests of other processors. The goal is to identify faulty processors given test results:

Huang, K., and T. Chen, "Three-Valued System Diagnosis," *Proc. 15th Int'l Symp. Multiple-Valued Logic*, May 1985, pp. 356-360.

Leonetti, R.A., and J.T. Butler, "Characterization of Diagnosability of Systems with Four-Valued Test Results," *Proc. 15th Int'l Symp. Multiple-Valued Logic*, May 1985, pp. 52-56.

Sengupta, A., and A. Sen, "Diagnosability of Systems with Three-Valued Test Outcomes," In *Computer Science and Multiple-Valued Logic: Theory and Applications*, second edition, D.C. Rine, ed., North-Holland, pp. 567-574, 1984.

Miscellaneous

Hayes, J.P., "Uncertainty, Energy, and Multiple-Valued Logic," *IEEE Trans. Computers*, Feb. 1986, pp. 107-114.

Papachristou, C.A., "Content-Addressable Memory Requirements for Multivalued Logic," *Proc. 11th Int'l Symp. Multiple-Valued Logic*, May 1981, pp. 62-72.

Rine, D.C., "A System Design Method from Logic Programming to Multiple-Valued Logic," *Int'l J. Electronics*, Aug. 1987, pp. 163-170.

* Silio, C.B., "Application of Multiple-Valued Logic to Microprogrammed Processors," *Proc. 11th Int'l Symp. Multiple-Valuded Logic*, May 1981, pp. 79-80.

* Smith, W.R. III, "Minimization of Multivalued Functions," In *Computer Science and Multiple-Valued Logic: Theory and Applications*, second edition, D.C. Rine, ed., North-Holland, New York, 1984, pp. 195-226.

Taking The Lead In Applied R&D

CONTEL·TECHNOLOGY·CENTER

"If there's a better way, we'll find it."

Dr. Alan Salisbury, Director
Contel Technology Center

It's been the key to our success for the past 25 years. We're Contel Corporation, and time and again we've managed to find better solutions to meet the telecommunications and information services needs of our customers. Today, we're a $3-billion Forbes 500 corporation. We're also the third-largest independent telephone company in the U.S. But we're not a company that stops with success. We're always trying to do better.

That's why we've taken the initiative with a brand new Technology Center responsible for the transfer of Applied R&D technology to our many products and services. In the following laboratories, we'll be turning emerging technologies into useful solutions. And bridging the gap between research and application:

- **Artificial Intelligence/ Man-Machine Interface**
- **Software Engineering**
- **Transmission and Switching Systems**
- **Networks and Secure Systems**

If you're ready to take the lead, and eager to turn *your* ideas into realities, the time is now. Because we're taking the lead. With the best minds in the business. Leading Scientists and Engineers who want to create practical solutions. Those who are looking for the freedom to design their own projects and influence our future direction. And, in our specially built, state-of-the-art facility in the Wohlstetter Technology Park in Fairfax County, Virginia, you'll be able to.

Close to the Nation's capital, you'll find Fairfax an upscale community offering affordable, quality housing, a shorter commute, superior educational systems, and a growing economy spurred by both commercial and residential development. Plus plenty of recreational and cultural choices. It's all here. It's time you were, too.

Please send resume, including salary history, to: Contel Technology Center, 12015 Lee Jackson Memorial Highway, Fairfax, Virginia 22033, (703) 359-7732. We are an equal opportunity employer m/f/h/v. U.S. Citizenship required.

CONTEL® Technology Center

Multiple-Valued Logic: A Tutorial and Appreciation

Kenneth C. Smith
University of Toronto

This tutorial is meant to intrigue and inform the reader whose interests and experience might not yet include multiple-valued logic. By raising and reviewing issues often of concern to those first encountering the topic, it sets the stage for the five articles on multiple-valued logic that follow. It assumes that the typical reader is broadly conversant with the modern field of (binary) digital computing and is at least generally aware of its mathematical foundations and constructs, as well as the nature of its important recent developments. Thus, this tutorial places the developments and potential of multiple-valued signals and logic in the relevant context of binary or two-valued signals.

This tutorial places the developments and potential of multiple-valued signals and logic in the relevant context of binary and two-valued signals.

Two-valued signals currently dominate the field of digital-computing machinery, but, historically, there is no lack of emphasis on radices (or bases) higher than two.[1] Base 10 has been important for all the obvious reasons, particularly in mechanical systems and in the users' view of electronic machines. Babbage's original design used 10-valued mechanisms, as did generations of later mechanical calculators. Modern electronic calculators present data in base 10 for user convenience, as did early machines such as the IBM 1620. With very few exceptions, such machines used inherently binary electronic circuitry and logic internally, with information encoded in binary-coded-decimal format, but often with computation done in (coded) decimal arithmetic.

Thus, the choice of radix may be optimized separately in either the conceptual or actual (that is, implementation) domain, and a "best" choice could be one where both optima coincide.

As we will explore shortly, issues of concern at the conceptual level include notation, operational description, and recognition of symmetry. At the actual level they include the use of physical space, noise margins in signal space, and conversion with binary.[2,3]

MVL's role in the binary world

Concerning the role of multiple-valued logic (MVL) and data representation in the dominantly two-valued or binary world, a multiplicity of views exists. First, it is unlikely that binary will capitulate or in any way wither and die. Besides its obvious entrenchment, there are very good reasons why two is a special value. For example, every designer of binary logic knows that a logic gate, say NAND or NOR, need have a fan-in of only two to be perfectly general, although not particularly flexible. He knows as well that, while large gate fan-out is very convenient, a fan-out of two is all that is necessary.

Thus, even from a binary perspective, we see that two is enough but more can be better. This leads us to the first view presented in this tutorial—namely, that multivalued logic has at least the potential for enriching the current two-valued reality.

Conversion with binary. Because of the necessary coexistence with binary logic, radix conversion is a topic of immediate concern. To the practicing binary logic designer, the required converters represent no great challenge conceptually. Such multiple-valued-to-binary and binary-to-

0018-9162/88/0400-0017$01.00 © 1988 IEEE

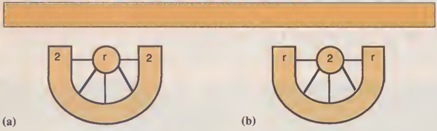


Figure 1. The digital universe: (a) binary view; (b) multiple-valued view.

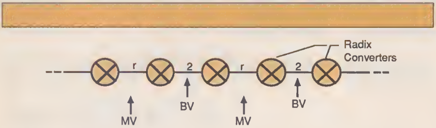


Figure 2. Another look at reality, described as a chain of two-valued and *r*-valued parts from two viewpoints: BV, binary-valued, and MV, multiple-valued.

multiple-valued converters are merely particular cases of the analog-to-digital and digital-to-analog converters with which the designer is already familiar. In particular and on the positive side, the required converters have very low precision by present-day analog standards and, accordingly, can be quite simple structurally. Fortunately, this simplicity is quite consistent with the obvious desire for speed, since in most cases (but not all!) too great a speed penalty might overwhelm the other advantages that multiple-valued data representation and logic can provide.

On the issue of conversion, a moment's reflection suggests that some particular choices of value in a multiple-valued system may be more attractive than others.

A special choice of radix—a power of two. Conversion with binary is most efficient with special choices of radix, ones in which no information is lost or left unused. These radices are exact powers of two, that is 2^n, for $n = 1, 2$, etc., of which obviously two itself, four, possibly eight, and potentially 16 show considerable promise. For example, as indicated later, in the area of memory-system applications, there have been commercial developments in four-valued ROM and proposals for even 16-valued RAM systems.

Two views of multiple-valued systems. As already noted, present and potential multiple-valued systems must be viable in the binary real world. Two such views are expressed symbolically in Figure 1. In each, the focus of attention is at the top and center—"on the ball," so to speak—and emanates from a related universe "U." As in many ideological issues, only the labels are different.

Figure 1's views can be combined and/or extended (and certainly detailed) as indicated in Figure 2. A modern view of digital reality, it shows a collection of interacting parts as labeled links coupled by converters (marked "x"). For simplicity, it reduces the obvious potential generality of the interconnection to only a single dimension. Also for simplicity, it indicates only radix 2 and radix *r* sections, although hybrids using a multiplicity of radices are clearly possible.

From Figure 2, one can appreciate the nature of two recurring issues in the multiple-valued literature (and related detailed views). They are represented pictorially by the arrow pairs BV and MV. BV, for binary view, presents the system as primarily binary, but possibly consisting of linkages that may be internally multivalued. The multivalued ROM alluded to earlier, and described later, exemplifies such a system, one in which the user's view is clearly binary, while the reality of its multivalued implementation may remain unrevealed or even concealed. Arithmetic processors constitute another important example, where multiple-valued data representation provides additional conceptual and implementational flexibility. Alternatively, MV, for multiple-valued view, presents the system as multiple-valued or as having multiple-valued segments, which in turn (internally) may have binary parts.

This idea appears in practice in at least two distinguishable ways. It's one way of constructing some multiple-valued gates, as well as a way for understanding many others. Quite separately, it describes a general scheme for using multiple-valued linkages to reduce wiring complexity between otherwise binary subsystems, in either VLSI or larger scale environments.

Later in this tutorial, in sections on application and circuit implementation, I discuss these issues further.

Representation

Extending systems from binary to radix *r*—that is, from the usual two-valued to a multiple-valued representation—requires choices involving the relative separation and ordering of values, issues (both conceptual and actual) that are less visible in two-valued systems.

While other possibilities exist, one usually thinks of the multiple values as *monotonic*, being separated in an established sequence at (nominally) equal intervals. As a consequence, just as in binary, a conceptual notion of relative magnitude arises, one extreme magnitude being larger (or smaller) than the other. Notationally, one extremum can be called zero and the other one, with intervening values represented by fractions.

Thus, a three-valued system (also called a ternary system) exists for which the values are called (0, ½, 1). Just as in binary, the "largest" value is considered to be one (the unit value, unity, u, "all there is," etc.). However, just as with "negative logic" in radix 2, the actual value of the physical variable corresponding to one need not be the largest.

Other notations exist as well. For an odd radix such as 3, the balanced ternary (also called "signed binary"!) notation (-1, 0, + 1) is often used to emphasize symmetry. More generally for radix r, the values can be labeled (0, 1, 2, . . . , (r – 1)), the corresponding three-valued notation being (0, 1, 2). The latter style of notation (and the name n-valued rather than n-ary) is emphasized in the present description.

Binary-related radices

As already noted, power-of-two radices ($r = 2^k$, k integer) have an obvious special relationship to the conventional binary system. In addition to the fact that conversion to and from binary is particularly convenient, the conversion is quite "efficient," since the information capacity of a single r-valued digit is exactly matched by that of k binary digits (where $r = 2^k$). This "direct-packing" property of binary-related radices is quite important in some of the more obvious applications of multivalued coding. Such is the case, for example, with the application of four-valued digit encoding (called quaternary and for which $k = 2$) in several commercial ROM designs and CCD designs described shortly. More recently, development of a 16-valued RAM design, for which $k = 4$, has been reported, although a product is not yet commercially available. Moreover, the success of a recently

Multiple-valued functions of one variable

The number and naming of one-variable functions is a non-issue in binary, but such is not the case in multiple-valued logic systems. The reason is a combinatorial one: an r-valued system has r possible outputs for each of r possible input states and, accordingly, r^r functions of a single r-valued variable. This corresponds to 2^2 or 4 possible functions in binary—namely, the identity, the complement, and the constants 0 and 1. Thus, in binary, only half the available functions—identity and complement—are really "active" or "interesting," and little more can be said.

But a multiple-valued radix-4 system, for example, has 256 such functions, only four of which are constant and 252 of which need names. Clearly, even in a four-valued system, the process of naming can get out of hand.

Nevertheless, for a variety of reasons related to our binary heritage, some special functions have been identified and labeled. The accompanying table views some that can be defined independently of the particular radix r. However, for illustration, examples in both radices 2 and 4 are provided.

Included also is a general-purpose functional notation. It consists simply of a string of r r-valued digits delimited by the symbols < >. In this string, each item is the output value produced by the corresponding input in the standardized reference input string (0,1,2, ..., r–1). For example, in the four-valued case illustrated, the identity function is labeled <0,1,2,3>, indicating the output is the same as the input for each value.

Selected functions of a single multiple-valued variable with four-valued and two-valued examples.

Ref. no.	Function name f	Preferred notation	Function definition $x_0 = f(x_i)$	Other notation	Four-valued example Function	Four-valued example Output	Two-valued example Function	Two-valued example Output	Binary equivalent
1	Identity	x	x		x	<0,1,2,3>	x	<0,1>	Identity
2	Complement	$\bar{x}$	$r-1-x$		$\bar{x}$	<3,2,1,0>	$\bar{x}$	<1,0>	Complement
3	Successor	$\overrightarrow{x}$	$(x+1)_{\text{mod} r}$	suc(x) x_{+1}	$\overrightarrow{x}$	<1,2,3,0>	$\overrightarrow{x}$	<1,0>	Complement
4	Predecessor	$\overleftarrow{x}$	$(x-1)_{\text{mod} r}$	x_{-1}	$\overleftarrow{x}$	<3,0,1,2>	$\overleftarrow{x}$	<1,0>	Complement
5	Cycle	x_k	$(x+k)_{\text{mod} r}$ $\{-r+1 \leq k \leq r-1\}$	$\overrightarrow{x}^k$	x_1 x_{-3}	<1,2,3,0> <1,2,3,0>	x_0 x_1	<0,1> <1,0>	Identity Complement
6	Literal	$\overset{a,b}{x}$	$r-1$ if $a \leq x \leq b$ 0 if $a > x > b$		$\overset{1,2}{x}$ $\overset{0,3}{x}$	<0,3,3,0> <3,3,3,3>	$\overset{0,1}{x}$ $\overset{0,0}{x}$ $\overset{1,1}{x}$	<1,1> <1,0> <0,1>	Constant Complement Identity
7	Universal literal	x^S	$r-1$ if $x = s \in S$ 0 if $x \neq s \in S$		$x^{1,2,3}$ x^1	<0,3,3,3> <0,3,0,0>	x^0 x^1	<1,0> <0,1>	Complement Identity

Table 1. The number of functions of one variable in radix *r*.

r	r^r
2	4
3	27
4	256
5	3,125
6	46,656
7	823,543
8	16,777,216

developed quaternary multiplier can be at least partly ascribed to this direct-packing property.[4]

In view of these facts, many of the examples presented in this tutorial are four-valued.

Multiple-valued functions

Functions of one variable. One can appreciate both the related power and potential complexity of multiple-valued systems by considering functions of only one variable. In general, for a single-variable function in radix r, r possible outputs are available for each of the r possible input values. Accordingly, there are r^r such one-variable functions. Thus, for the binary case, where $r = 2$, there are $2^2 = 4$ possible functions of a single variable x_i, namely, for $x_i = 0$ or $x_i = 1$, the output x_o can be 0 or 1. That is, the output can be the same as the input ($x_o = x_i$, an identity function), reversed from the input ($x_o = \bar{x}_i$, the binary complement function) or constant (either 0 or 1). As the radix increases, so does the number of functions—quite dramatically, as shown in Table 1, where in radix 4, for example, the number of single-variable functions is seen to be 256.

The consequence of this increase is that the usual binary practice of simply naming the few available functions (that is, identity, complement, and constant 0 or 1) gets out of hand in higher-radix systems. Thus, while some named functions prevail, a more general approach is needed. One approach is simply to define the function as an output r-tuple, a simple list (or string) of outputs corresponding on a one-to-one basis to inputs in the usual (monotonic) ascending order.

Consider, by way of example, a four-valued variable for which the reference input string is (0, 1, 2, 3). The corresponding identity output function is straightforward and is written as $<0, 1, 2, 3>$. In this notation, $< >$ signifies an output string whose members are in one-to-one correspondence with those of the reference input string. Thus, for the identity function, 0 input produces 0 output, 1 input produces 1 output, and so on. Correspondingly, a particular complement, written $<3, 2, 1, 0>$ and for which, in general, $x_o = \bar{x}_i = r-1-x_i$, is often called diametrical negation or inversion. But what about all the others? Clearly there are a lot of possibilities (in fact, 254 remaining combinations), and simple names are quickly exhausted.

Historically, there are several specially named multiple-valued functions of a single variable. As well as the (diametrical) inverter already described, these include the "cycle" and the "literal" functions. The *cycle*, for which $x_o = \overrightarrow{x}_i = (x_i + 1)_{\text{mod} r}$ in general (where in modular arithmetic addition a result $\geq r$ is reduced by r) becomes $<1, 2, 3, 0>$ in the particular four-valued example. A variation, called a downward or negative cycle ($\overleftarrow{x}_i$), in which addition is replaced by subtraction, becomes $<3, 0, 1, 2>$ in the radix-4 example. Cycling by a value k, rather than simply by 1, is also possible.

The *literal*, for which $x_o = {}^a x_i^b = r - 1$ if $a \leq x_i \leq b$ and $x_o = 0$ otherwise, would become $<0, 3, 3, 0>$ in the four-valued case in which $a = 1$ and $b = 2$. When $a = b$, a *simple literal* ($x_i^a = {}^a x_i^a$) results, becoming $<0, 3, 0, 0>$ in the four-valued case in which $a = 1$. This particular idea has been extended to provide a *universal literal*, for which $x_o = x_i^{S_i} = r - 1$ if x_i takes a value a_i in the set S_i and $x_o = 0$ otherwise. Thus, in a four-valued case where S_i is (0, 2), $x_i^{0,2}$ becomes $<3,0,3,0>$. (The reader will come to appreciate the true notational power of the universal literal when a discussion of issues related to implementation is reached.)

In addition, there are other special functions identified historically for a variety of reasons, including convenience of application, analysis, and implementation. Generally, in hindsight, they can be described in terms of the universal literal notation. For a summary of some of the notable functions of a single variable, see the table in the sidebar titled "Multiple-Valued Functions of One Variable." Figure 3 shows transfer graphs or transfer characteristics of some of the more interesting and useful functions of one variable, namely, the identity, complement, single-threshold inverting, and cycling functions. Note that this figure shows the input switching thresholds midway between each of the input values.

Functions of more than one variable. In the discussion of functions of a single variable, we have already seen the conceptual and notational utility of two simple concepts, that of the string notation and input-output transfer graphs. These ideas—although available—are rarely employed in binary for two reasons: first, because their use with two values is somewhat degenerate and, second, because convenient higher level functional notations are available.

In contrast, for two or more variables, the tabular and mapping techniques common to binary are used directly for multiple-valued design. The only important difference is one of size, obviously related to the increased number of values to be tabulated. For example, the truth table for a two-variable four-valued two-output function (the arithmetic sum and carry) shown in Table 2 has $4^2 = 16$ rows. Obviously, however, four binary digits are needed for the same input information content embodied in two radix-4 digits, with the corresponding table also having $2^4 = 16$ rows. Figure 4 shows a two-variable four-valued map for the arithmetic sum function.

Thus, for the purpose of reducing size alone, if for no other reason, there is an important need for a multiple-valued functional notation.[3]

For generality in describing functions of two or more multiple-valued variables, consider x_i to be such a variable with a value from the set $V = \{0, 1, 2, 3, \ldots, r-1\}$ and $X = \{x_1, x_2, \ldots, x_n\}$, the set of n such variables. However, for expediency, examples will be posed in terms of two variables in a four-valued system, that is for $n = 2$ and $r = 4$.

While the number of functions of even two r-valued variables can be quite large (r^{r^2} in general) in comparison to the two-valued case (where $2^{2^2} = 16$), fortunately, the number of basic common constructs does not grow correspondingly. Note also that for the particular case of $r = 2$, each directly reduces to a familiar binary form.

Historically, the most important multiple-valued connectives have been the *maximum* (MAX) and *minimum* (MIN) functions, whose outputs are respectively the largest and the smallest of the n values presented. Note that for two-valued variables, *maximum* corresponds to the binary OR while *minimum* corresponds to AND. This correspondence justifies the following notation, where for multiple-valued variables x_i, $1 \leq i \leq n$,

$$\text{MAX}(x_1, x_2, \ldots, x_n) = x_1 \vee x_2 \vee \ldots x_n, \text{ is the largest } x_i, \text{ and}$$
$$\text{MIN}(x_1, x_2, \ldots, x_n) = x_1 \cdot x_2 \cdot \ldots x_n, \text{ is the smallest } x_i.$$

Corresponding as well to the binary tradition is the notion of *sums* and *products*, with the *sum of products* and *product of sums* as general functional constructs.

In addition, however, motivated by the natural association of multiple-valued variables with arithmetic, there are other, more arithmetic, connectives including arithmetic sum, difference, and product forms. Because of their relatively simple electronic implementation, only sum and difference forms will be considered in this limited tutorial. Note that the arithmetic-product forms, though not considered here, may assume greater importance with the potential application of multiple-valued techniques to optical signal processing. Thus, besides MAX and MIN, there are TSUM and MODSUM operations, which are truncated and modular summation,[5] respectively.

Accordingly, for a two-variable, four-valued environment,

$$x_1 \cdot x_2 = \text{MIN}(x_1, x_2)$$
$$x_1 \vee x_2 = \text{MAX}(x_1, x_2)$$
$$x_1 + x_2 = \text{SUM}(x_1, x_2) = (x_1 + x_2)$$
$$x \boxplus x_2 = \text{TSUM}(x_1, x_2) = \text{MIN}(x_1 + x_2, 3)$$
$$x_1 \oplus x_2 = \text{MODSUM}(x_1, x_2) = (x_1 + x_2)_{\text{mod}4}$$

where the symbolism is

- $\cdot$ for MIN (AND)
- $\vee$ for MAX (OR)
- $+$ for arithmetic addition (plus)
- $\boxplus$ for truncated addition
- $\oplus$ for modulo-r addition (Exclusive-OR, or EXOR, in binary)

Generally speaking, which connective is most appropriate depends on the technology available and the application's need for minimization. Thus, in some technol-

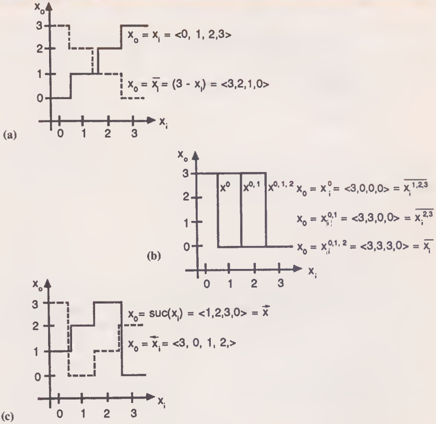
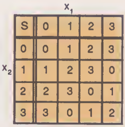


Figure 3. Transfer characteristics of sample functions of a single four-valued variable x: (a) identity x and complement ($\bar{x}$); (b) single-threshold inversion; (c) successor (suc(x)) and cycle ($\vec{x}$, $\overleftarrow{x}$).

Table 2. A truth table for the sum (S) and carry (C) outputs of a half adder for two four-valued inputs (x_1, x_2)

x_1	x_2	S	C
0	0	0	0
0	1	1	0
0	2	2	0
0	3	3	0
1	0	1	0
1	1	2	0
1	2	3	0
1	3	0	1
2	0	2	0
2	1	3	0
2	2	0	1
2	3	1	1
3	0	3	0
3	1	0	1
3	2	1	1
3	3	2	1



S	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Figure 4. A map of the sum (S) output function of a four-valued half adder having two inputs (x_1, x_2).

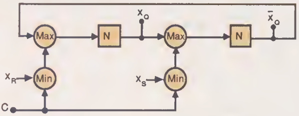


Figure 5. A clocked multiple-valued storage element using complement (N), MAX, and MIN gates.

ogies (notably those using current or charge), summation operations are very convenient. However, the corresponding complexities of don't-care conditions make minimization relatively difficult. Conversely, in these same technologies, MAX and MIN are somewhat complex, but don't-care conditions are straightforward and highly visible, making minimization relatively easy.

Storage techniques in MVL

Techniques for storage of digital information in multiple-valued logic systems relate directly to those commonly encountered in binary systems. The major difference is that the information represented by a signal, and stored within a single multistable element, can be much greater than in the binary case. However, in both the binary and multiple-valued worlds, gating (that is, the decision of whether to renew or retain the information stored) remains an essentially binary function.

In both binary and multiple-valued systems, two types of storage mechanism are normally encountered. The most economical scheme in binary, namely, dynamic storage using a capacitor, becomes particularly attractive with multiple-valued signals since a single capacitor with a simple transmission gate has the potential for storing much more information.[6] Practical problems, generally related to noise margins in readout and refresh operations, abound in such systems. Nevertheless, as far as the storage medium and simple gating alone are concerned, the storage efficiency of multiple values on a single capacitor is certainly attractive. This is particularly the case in hybrid implementations using a combination of dynamic and static storage for economy and speed—for example, in master-slave register-element construction.

Structures for static storage in multiple-valued systems generally follow the pattern set in the binary case, namely, that of an active positive-feedback loop. For example, just as in binary, a feedback loop consisting of two inverters or complement circuits (that is, gates for which $x_o = \bar{x}_i$) will exhibit stable states, in particular r of them in a radix-r system. Gating is also similar in multiple-valued and binary schemes, with MAX and MIN replacing OR and AND. A clocked SET-RESET flipflop following this general approach is shown in Figure 5.

In Figure 5, N represents a complement circuit. In a four-valued environment, its function $\bar{x}$ is described by <3,2,1,0> (that is, the output corresponding to the input string (0,1,2,3)). All signals x are multiple-valued; signal C is a two-valued clock that alternates between the values of zero and three. Signals x_R and x_S represent multiple-valued Reset and Set inputs, respectively, which are usually (but not necessarily) complementary. Signals x_Q and $\bar{x}_Q$ are the (usually) complementary multiple-valued outputs of the four flop. When the clock C is low (at zero), the lower input of each of the MAX gates is held low, and the storage state prevails. When the clock is high (at three), the four-flop state is changed to a value established by x_R and x_S. For $x_R = \bar{x}_S$, $x_Q = \bar{x}_Q = x_S$ once C has fallen to zero. Note that the static storage element shown in Figure 5, is "signal-level-restoring," that is, has correctly quantized output-signal values, by virtue of the design of the complement circuits.

Implementation

Basic issues. While there is much to be said about implementation, and much of that is explored in other articles in this special issue of *Computer*, it is likely of interest here, for the general reader, to explore some of the underlying issues.

First, let us recognize once again that the fundamental abundance of binary devices, originally mechanical and now electronic, underlies the current dominance of binary technology. The early mechanical age of computation saw a recurring appearance of multiple-valued techniques, for example, detented cogs having 10 stable positions. As fortune would have it, however, multistate electronic devices are far rarer.[3] Whether this is good or bad, time alone will tell.

But note two things in passing. First, while there is a lack of basic electronic multistate devices, relatively simple equivalent constructs of analog and binary devices are available. Second, the history of technology displays various recurring cycles; thus, it is not unreasonable to expect that some emerging technology may have more multivalued attributes. This is, for example, a possibility with electro-optical and optical technologies. Meanwhile, while waiting, many real developments can be implemented *now*. Expanding on this possibility is the direct motivation of this section and of other articles in this issue of *Computer*.

Signal representation. In electronic implementations of binary digital systems, it is evident that signals can be (and are) represented directly by voltage, current, or charge or by some combination. In fact, the interrelationship of these variables through the capacitive and resistive elements of a VLSI environment almost makes such distinctions irrelevant. Nevertheless, circumstances can bias one's deliberations in binary design (and in multiple-valued design for that matter). The existence of oscilloscopes, which measure voltage so easily, tends to bias thinking and descriptions (such as waveform presentations) in the voltage direction. Also, some technologies are easily described in one domain but less well in

A universal multiple-valued building block: The T-gate

Overview. As an aid to understanding the implementation aspects of multiple-valued logic functions, let's consider, at a general functional level, a particular "universal" element called the T-gate.[1,2]

The T-gate qualifies as a universal element in several different senses. First, it is logically complete; that is, all multiple-valued functions of one or more variables can be created with a T-gate. Second, its operation is intuitive; that is, one can relatively easily understand what it does, both as a single basic element and in collections. Third, it is easily implemented; that is, its construction in most, if not all, available technologies is quite straightforward. Fourth, it highlights two essential elements that must be embodied in any logic gate, namely, logic-value thresholding and logic-signal connection or switching.

But don't assume that a T-gate constitutes the ultimate in multiple-valued constructs. It has some shortcomings: While intuitive, the T-gate does not lead to a very effective scheme for minimization. While usable as both a model for all multiple-valued realizations and a means to their actual implementation, the T-gate's models and implementations are seldom minimal. Thus, for multiple-valued logic, the T-gate is a useful general-purpose tool; but like other general-purpose tools, it does everything to a degree but few things particularly well.

In terms familiar to a binary designer, the T-gate is simply a multiplexor. It is, however, specially adapted to an r-valued world. In particular, it has $r+1$ inputs, one of which is an r-valued control input whose value determines which of the other r (r-valued) inputs is selected for output.

The idea is shown conceptually in the figure at left below, where x_k is a four-valued controlling or selecting input that takes on values (0,1,2,3). For $x_k=0$, input x_1 is selected; for $x_k=1$, x_2 is selected, and so on. Thus, in general $x_0 = \langle x1, x2, x3, x4 \rangle$, using the string-sequence notation introduced in the main text.

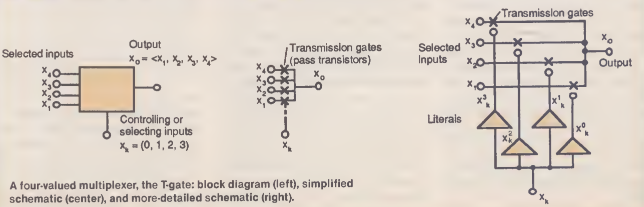
The middle figure below, a schematic internal view of the T-gate, distinguishes the separate roles of the selected and selecting inputs. Here, the "x" symbols denote pass-transistor switches, or transmission gates, controlled by mutually exclusive binary signals under control of the selecting input x_k. The figure at right below is a more detailed schematic in which one-at-a-time operation of the r pass transistors is more precisely specified by use of four universal-literal gates driven by the selecting input x_k.

As usual with r-valued constructs, the two-valued T-gate is quite recognizable; in fact, it is normally described as a transmission gate. Incidentally, the identification of it as a general-purpose transmission gate, in which the controlled input and output are not restricted to two-valued signals and may in fact be analog, foreshadows its more-general multiple-valued application. This is exemplified in the sidebar on T-gate applications (see p. 26).

Related insights. Now that we've viewed some overall aspects of the T-gate, let's reflect on the insights it provides to general implementation problems.

As noted, the T-gate involves threshold detection, an inherently binary process. The thresholds, as the figures indicate, are relatively complex; however, being double, as the universal-literal notation in the right-hand figure shows, the switch-controlling outputs are still binary. In this sense, the T-gate follows the general scheme introduced as MV in the main text (Figure 2), where multiple-valued signals are first decoded (by thresholding) and then processed using binary techniques. On the other hand (a hint of a richer world), although T-gate switches are obviously under binary-signal control, signals in the controlled path can be more general. In particular, they can be multiple-valued.

This alone can be an interesting observation for those of binary persuasion, in view of the current importance of pass-transistor logic in binary VLSI. It is, perhaps, even more intriguing to reflect on two other associated facts. First, the passpath, often of considerable length in some VLSICs, if made multiple-valued, can pack more information.[3] Second, the pass-transistor-controlling signals, while still binary and therefore relatively numerous, are typically located in the immediate vicinity of the threshold detectors, and they can be quite short.



A four-valued multiplexer, the T-gate: block diagram (left), simplified schematic (center), and more-detailed schematic (right).

1. M. Kameyama, T. Hanyu, and T. Higuchi, "Design and Implementation of Quaternary nMOS Integrated Circuits for Pipelined Image Processing," *IEEE J. Solid-State Circuits*, Feb. 1987, pp. 20-27.
2. T. Higuchi and M. Kameyama, "Ternary Logic System Based on the T-Gate," *Proc. Fifth Int'l Symp. Multiple-Valued Logic*, Bloomington, Ind., May, 1985, pp. 290-304.
3. K. Horie et al., "Design of a Complementary Pass Gate Network for a Multiple-Valued Logic System," *Proc. IEEE 17th Int'l Symp. Multiple-Valued Logic*, Boston, May 1987, pp. 142-149.

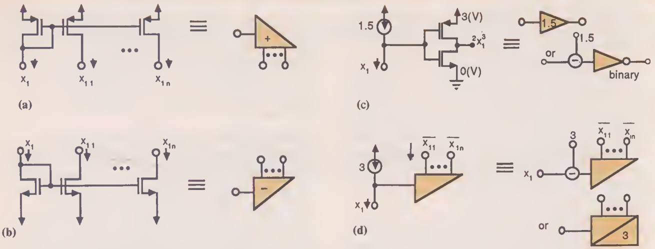


Figure 6. Component circuits of a simple current-mode four-valued circuit family: (a) positive replicator, (b) negative replicator, (c) literal ${}^2x^3$, (d) complement $\bar{x}$ with replicated outputs. In the circuits shown, the reference current values are conceptual. In practice, a small fixed current bias would be added to ensure that the mirrors are never cut off. This is a technique called "fat zero" in other, but related, contexts.

others. But in general, in the binary world, such distinctions do not matter; one uses the most convenient description, one which usually dwells on the assumed reality of zero and one, independent of implementation.

However, for multiple-valued systems in VLSI, it is no longer appropriate to ignore these distinctions, particularly between voltage and current representations. The reason is that each may be seen to impact directly (and dominantly) on one of two critical dimensions of the VLSI defining domain.

The fact is that a voltage representation impacts directly on the supply-voltage dimension of VLSI. In binary systems, design details affect noise margins in subtle ways, but it is the value of the power supply that really sets the tone. Witness the controversy currently surrounding the choice of a standard supply voltage for VLSI design. While five volts was once necessary for many reasons, it is now clearly far greater than necessary for many (perhaps most) applications and newer technologies, and it wastes considerable power and even reduces speed. In fact, other standards exist, certainly for portable applications such as watches, within VLSI application-specific and custom chips.

This issue is critical when considering a multiple-valued-voltage representation. It could even be argued that straightforward multiple-valued-voltage representation is feasible only because of the voltage surplus in most present-day binary systems.

On the other hand, true multiple-valued-current logic does not require increased voltage space (although current-voltage hybrids do). Rather, to maintain the particular device-current density that characterizes a particular VLSI process, device width must increase in proportion to the multiplicity of current values to be used.

The above discussion has highlighted a distinction, once ignored in binary, that is becoming important: multiple-valued signaling needs space. Whether this is in the voltage-supply dimension or in a linear chip dimension is an important design choice. The voltage choice, if taken, tends to be system-wide, while the current choice is more localized, affecting only the part of a VLSI system in which multiple-valued generating (and detecting) circuits exist. A positive aspect is that both are linear with radix.

This tutorial concentrates on multiple-valued signals in the current domain. Occasionally, however, hybrid systems will be referenced, ones in which a voltage representation is used quite locally, where perhaps smaller noise margins, and accordingly less voltage span, can be tolerated. Note that charge-based systems are also hybrid in a related sense, namely, that an inevitable variation of voltage exists across a fixed capacitor (or charge well). In a practical sense, however, such voltages can be small, since relevant connections are local. The common (but not essential) need to convert signals to the multiple-valued voltage domain is, of course, accompanied by the obvious problems associated with limited voltage space.

A current view. In any logic implementation, a major consequence of the voltage/current duality is that difficult voltage-mode operations become easy in current mode and (regrettably) vice versa. Thus, for a TTL NAND operating in voltage mode, fan-out is a straightforward wiring task, but fan-in requires some particular hardware for each input. In current mode, the reverse is true: fan-in is very simple by wire connection, but fan-out is much more complex, requiring circuitry of the type shown in Figures 6a and 6b. For this reason, current-mode binary circuitry was quite uncommon until creation of *binary I^2L* (integrated injection logic) recognized the convenient implementation and utility of multicollector transistors. The realization by Dao[7] and others of the relationship of these structures to current mirrors led to the development of *multiple-valued I^2L* in the 1970s. This was the first use of mirrors as signal-

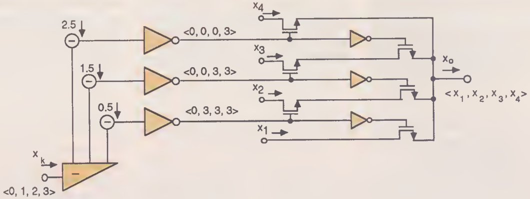


Figure 7. A T-gate implemented in a simple current-mode four-valued circuit family. The switches shown as nMOS are appropriate for this application. They are easily converted to CMOS by employing the existing (binary) CMOS inverters (shown as triangles) to drive the additional pMOS pass transistors.

current replicators in digital systems.

Component circuits of a simple current-mode multiple-valued circuit family are shown in Figures 6a-d and 7. Simplistically viewed, the scheme is radix-independent. However, for practical reasons related primarily to mirror imperfections, the radix is limited. Certainly, four-valued designs have been demonstrated and will be considered by way of example.

Figures 6a and 6b show replicators using simple mirrors to provide *n* copies of a multiple-valued current signal x_1. Figure 6c, the circuit of a particular literal, $x^{2,3}$, illustrates two general ideas: arithmetic subtraction by wiring and use of the (binary) sign of the result to operate a simple binary switch. In Figure 6d, arithmetic subtraction and mirroring are combined to produce *n* copies of the complement $\bar{x}_1$. These examples are simple functions of one variable; other multivariable constructs are available by wire connection at the input.

Figure 7's T-gate schematic illustrates the scheme's potential generality, its logical completeness and extension to multiple variables. The figure adds only a transmission gate (or pass-transistor switch) to the elements introduced in Figure 6. A simple nMOS version is shown for its simplicity and because most applications can tolerate its imperfections.

An applications overview

Memory. One obvious virtue of multiple-valued data representation is its potential for reducing the number of lines required for the parallel transmission of large amounts of data. The intense need for compaction in memory arrays has led to several commercial memory developments using multivalued data coding.[8]

Most prominent among these are several commercial ROM structures. Intel has used four-valued ROMs in two commercial products—the 8087 arithmetic coprocessor and a 432-03 peripheral-processor product. In each case, two-bit encoding reduced ROM array size enough to significantly increase array product viability. Motorola and General Instrument also report using four-valued ROM in somewhat more specialized applications—for example, driving a speech-synthesizer chip in an electronic toy.

These implementations use two distinct approaches. Intel's designs use a conventional IC process with four different channel widths for the ROM-cell MOS device. Even though the cell site must be larger to accommodate the largest transistor (conceptually three times as wide as in the binary case), the reduction by two in the number of cell sites and data lines produced an important reduction in array size. While the detection circuitry for four values of sense current is more complex, its inherent sharing in a memory environment saves considerable space overall.

On the other hand, in one of the Motorola designs and in the General Instrument designs, process complexity was increased to allow multithreshold MOS devices. Since the voltage dimension rather than a spatial dimension was used for coding, the resulting array was conceptually as small as a regular binary one. Furthermore, because of the relatively low speed of the application, General Instrument was able to use a very simple voltage-level scanning technique, which was slow but very economical of physical space.

Concerning other (more dynamic) applications of multiple-valued coding in memory, two CCD four-valued serial memory designs have been reported, one by IBM and one by Mitsubishi.[3] The general idea, multiple values of charge in a dynamic memory environment, has been extended in a pre-commercial study of a 16-valued RAM prototype reported at ISSCC by Hitachi.[6]

Communications. Specialists in relatively long-distance communications have long recognized the potential for information compaction and/or bandwidth reduction of nonbinary digital-signal transmission. Thus, communications designers evolved, quite separately from the development of multiple-valued logic, a signaling form they call multilevel coding. The ugly realities of transmission media make received signals less than ideal and detection a messy process, but potential exists for application of multiple-valued techniques in the pre- and post-processing of signals.

Concerning commercial communications applications of a somewhat less-global scale, at least one continuing product, introduced by Motorola in 1980, comes to mind. The MC 145026/7/8 is a customizable family of CMOS chip pairs used for remote-control applications. Data is transmitted between the chips as a serial stream in which information is encoded in four-valued units consisting of two binary digits. However, implementing the required reliability in self-clocking uses only three of the four values. Each chip has nine pins for data and/or address, which

T-gate applications

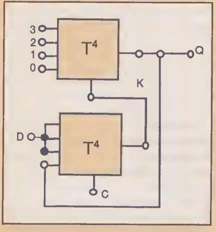
Single-variable functions. The T-gate, being a multiplexor under the control of the *r*-valued selecting input, can be used directly to implement any function of a single *r*-valued variable connected to its selecting input. For this purpose, the selected input corresponding to each selecting-input value is connected to a reference value appropriate to the output required by the desired single-variable function. Some four-valued examples are provided in the table below.

The idea can be extended to the formation of sequential machines. The figure at top right shows an arrangement that implements a simple four-valued D-type multiflop. Notice that a four-valued T-gate is used for gating, the feedback loop being closed when *C* is zero and the *D* input selected when *C* is one (or two or three). In this application, the upper T-gate functions as a four-valued quantizer for signal restoration.

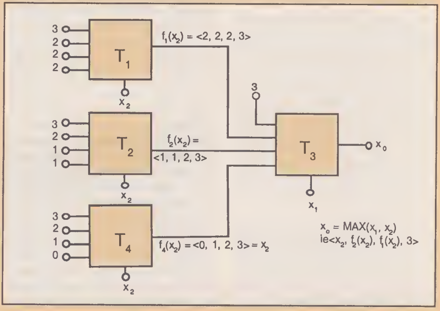
Multiple-variable functions. All functions of two or more variables can be constructed from T-gates. For example, the figure at bottom right presents a direct implementation of a four-valued MAX function of two variables using T-gates. Its output value is always the larger of two input values presented. The implementation is intended to correspond to a two-variable truth table in which x_1 is tabulated as the more global variable. Note that the fourth gate (T_4) is used only to provide a completely level-restored output, since only the controlling or selecting input of a T-gate is inherently level-restoring; for the other selected inputs, any input imperfection is propagated and (depending on implementation specifics) is possibly contaminated further.

Using the T-gate to implement all functions of one variable—selected examples.

x_1	x_2	x_3	x_4	$x_0 = f(x_k)$	Function	Symbol
0	1	2	3	<0,1,2,3>	Restoration	x_k
3	2	1	0	<3,2,1,0>	Complement	$\overline{x_k}$
3	0	3	0	<3,0,3,0>	Literal	x_k^{02}
2	2	2	2	<2,2,2,2>	Constant	2
0	0	3	3	<0,0,3,3>	Threshold	x_k^{23}



A simple D-type multiflop using T-gates, where T^r denotes a T-gate having an *r*-valued control input. Note that signals D, Q, and K are four-valued, while C is a two-valued clock in this four-valued example.



A T-gate implementation of the MAX function of two variables in a four-valued logic system.

can be used conventionally in a binary mode, but which also function in a ternary mode if desired. An open condition signals the third state (½) for input and output, the latter being the conventional tristate mode, well-known to binary designers. While the feature is quite simple and optional, the result is quite spectacular. Using multiple-valued coding can extend the chip pair's addressing range from $2^9 = 512$ to $3^9 = 19{,}683$, a dramatic improvement in a low-cost, pin-limited DIP environment.

Note that the tristate feature mentioned above, while ensconced unquestioningly in the binary world, is a multiple-valued idea in several senses. Moreover, in terms of the voltage versus current views of logic signaling described earlier, Motorola applies it in an "intimate" hybrid sense: the outer levels are voltage-driven by a gate output, while the inner (middle) level is current-driven by the gate output, with the corresponding voltage defined by the receiving input.

Testing. Three related areas of binary design needing more than a binary description are digital-signal dynamics, testing, and fault analysis. In response to this long-recognized need, a body of work has grown in which multiple-valued techniques play a part. Initially, the role was simply notational, for example, making explicit the transient (or otherwise undefined) state between zero and one. Gradually, more intervening conditions needed explicit consideration. Finally, algebraic aspects of multivalued logic were brought to bear in providing more formal tools for presentation and computerization of relevant tools.

Recently, for example, Hayes[9] developed a logic-circuit description that accurately represents the structure and behavior of MOS circuits using a $(3n+1)$-valued pseudo-Boolean algebra, where $n \geq 1$ is the number of fault "strengths" (ranging from open to short) to be modeled.

Another potential MVL application is in designing multiple-valued circuits that, while used primarily in the binary mode, exhibit and propagate failures in real time by transition to a multiple-valued mode. This idea has been exhibited by Hu[10] in actual circuits, although admittedly in a far from commercially viable form.

Arithmetic. The existence of a natural relationship between arithmetic and multiple-valued logic is a relatively old idea, exhibited in several quite different ways. I've already hinted at one way in the commentary on current-mode circuits, where current-mode summation of multiple-valued signals (as of analog signals) becomes quite trivial through wiring. In a quite different direction, but also an old idea, is the notion of multiple-valued recoding of binary signals to eliminate accumulating carry-propagation times in arithmetic processes. One such scheme is signed-quaternary, a seven-valued variant on four-valued signaling using the values -3, -2, -1, 0, 1, 2, and 3.

Another interesting attribute of multivalued coding in arithmetic is that truncated odd-multivalued encodings (3,5, etc.) produce unbiased results; that is, the less-significant part abandoned is equally often larger or smaller than one in the last-retained place. In binary (and other even bases), the rounding operation to produce an unbiased result is much more complex. Finally, because arrays of basic elements are common in arithmetic applications, the potential for increased use of each connecting link using multiple-valued signaling is particularly attractive.[3,4]

Broadly stated, the goal of this tutorial has been to stir the nonspecialist reader's interest in multiple-valued representation and logic. It has not been a survey, for that is either a more extensive or more intensive task. Rather, it has attempted to put into context, particularly for one proficient in binary systems, some of the major issues enveloping the topic. Clearly, more could be said; of this, much is presented in the specific context of the following articles. ☐

Acknowledgment

This work has been supported by the Natural Sciences and Engineering Research Council of Canada through Operating Grant A1753.

References

1. D.C. Rine, ed., *Computer Science and Multiple-Valued Logic, Theory and Applications*, 2nd edition, North-Holland, Amsterdam, 1984.
2. K.C. Smith, "The Prospects for Multivalued Logic: A Technology and Applications View," *IEEE Trans. Computers*, Sept. 1981, pp. 619-634.
3. S.L. Hurst, "Multiple-Valued Logic—Its Status and Its Future," *IEEE Trans. Computers*, Dec. 1984, pp. 1160-1178.
4. S. Kawahito et al., "A High-Speed Compact Multiplier Based on Multiple-Valued Bidirectional Current-Mode Circuits," *Proc. IEEE 17th Int'l. Symp. Multiple-Valued Logic*, Boston, May 1987, pp. 172-180.
5. S.P. Onneweer and H.G. Kerkhoff, "High-Radix Current-Mode CMOS Circuits Based on the Truncated Difference Operator", *Proc. IEEE 17th Int'l Symp. Multiple-Valued Logic*, Boston, May 1987, pp. 188-195.
6. M. Aoki et al., "A 16-Level/Cell Dynamic Memory," *IEEE ISSCC Digest*, Feb. 1985, pp. 246-247.
7. T.T. Dao, "Threshold I^2L and Its Applications to Binary Symmetric Functions and Multivalued Logic," *IEEE J. Solid-State Circuits*, Oct. 1977, pp. 463-472.
8. *D.A. Rich, "Survey of Multivalued Memories," IEEE Trans. Computers*, Feb. 1986, pp. 99-106.
9. J.P. Hayes, "Pseudo-Boolean Logic Circuits," *IEEE Trans. Computers*, July 1986, pp. 602-612.
10. M. Hu, "A Novel Built-In Test Technique for CMOS VLSI Circuits," *Proc. IEEE 16th Int'l Symp. Multiple-Valued Logic*, Blacksburg, Va., May 1986, pp. 80-84.

Kenneth C. Smith is a professor in the Departments of Electrical Engineering, of Computer Science, and of Library and Information Science at the University of Toronto. His research interests are in linear and digital electronics, computer architecture, instrumentation, and human factors. He is the author or coauthor of well over 100 journal and symposium papers and a contributor to several books, including *Microelectronics Circuits*, second edition, coauthored with Adel S. Sedra. He is also an advisor to corporations in the US and Canada.

An IEEE fellow, Smith received his BA, MA, and PhD degrees in engineering science, electrical engineering, and physics, respectively, from the University of Toronto.

Smith's mailing address is Dept. of Electrical Engineering, University of Toronto, Toronto, Ontario, M5S 1A4, Canada.

Comparison of Binary and Multivalued ICs According to VLSI Criteria

Daniel Etiemble

Université P. et M. Curie (Paris 6), France

Michel Israël

Institut d'Informatique d'Entreprise (CNAM), France

Ten years ago, the question regarding multiple-valued circuits was fundamental: Is it possible to design *m*-valued integrated circuits? Over the past 15 years, many *m*-valued circuits have been presented. A good survey of early designs can be found in Smith[1] and Hurst.[2] Today, the obvious answer to the question is yes.

Since 1972, the *Proceedings of the International Symposium on Multiple-Valued Logic* have presented many papers on *m*-valued integrated circuits or *m*-valued circuits compatible with IC technologies. These designs use bipolar technology, such as integrated injection logic (I^2L) and emitter-coupled logic (ECL); complementary metal oxide semiconductor (CMOS) technology; n-type MOS (nMOS) technology; and charge-coupled device (CCD) technologies.

How these ICs differ in terms of speed, chip area, and power dissipation demonstrates their usefulness for differing applications.

At the same time, the ability of microelectronics to provide efficient two-valued devices and circuits has improved. This development is expressed in the Moore and Noyce law illustrated in Figure 1. In the figure, the size of MOS memories is shown as a function of years, together with relevant microprocessor chips. The figure indicates that the number of devices per unit area increases exponentially. This exponential increase will not be limited by physical considerations before 1995, at the earliest. In 1987, CMOS memories exceeded 1 million devices. By 1995, western European IC manufacturers expect up to 100 million devices for MOS memories, 10 million devices for MOS logic, 1 million devices for bipolar memories, and 100,000 devices for bipolar logic with two-valued integrated circuits.

The IC industry must solve many problems to maintain this exponential growth. The problems entail interconnections (both on-chip and between chips), packaging, and cooling. Routing of interconnections on chips is well known as a major problem, and the silicon area used for interconnections may be greater than that used for the active logic elements.

The use of circuits with more than two levels has been offered as a solution to these interconnection problems. However, IC manufacturers have not made extensive use of *m*-valued signaling. With microprocessor dual-in-line packages, time multiplexing is the solution most often used when pin limitation exists.

To solve the pin-limitation problem, specific packaging techniques with more than 200 pins have been developed; examples include a regular pin-grid-array package, and IBM's Thermo Conduction Module (TCM). These packages solve several problems: signal distribution minimizes delays in off-chip nets, power distribution extends the operational limits of the circuits, and cooling maintains an acceptable junction temperature for the sake of functionality and reliability.

On-chip, the number of metal layers is increased to solve interconnection problems. nMOS technology has one metal layer, CMOS technology has two metal layers, and bipolar technologies typically have three or four metal layers. Therefore,

0018-9162/88/0400-0028$01.00 © 1988 IEEE

suggesting multiple-valued circuits as a solution to the interconnection problem is insufficient. The assertion that m-valued circuits solve the problem must be proved.

It is possible to realize m-valued ICs. But the more significant question is: According to very large scale integration (VLSI) criteria, is it possible to design m-valued ICs with performance better than or equivalent to the performance of the corresponding two-valued ICs?

We need a serious and fair comparison between two-valued and m-valued ICs according to VLSI criteria. For IC manufacturers, the first VLSI criterion is money: silicon (chip area) is money; time (design time, that is) is money. Thus, performance of integrated circuits is a trade-off between several criteria: speed, power dissipation, chip area, yield, CAD programs for IC design, and so forth.

Since yield and cost strongly depend on process complexity, the technology should be as simple as possible. For instance, the basic characteristics of 1.2- or 1-micrometer channel-length CMOS processes are: only n and p enhancement-mode transistors, two metal layers, no depletion-mode transistors, and no buried contacts. More complicated technologies using several implants for several threshold voltages, more polysilicon, or metal layers will only be used if they lead to significant improvements in the trade-off of speed, chip area, and power dissipation.

In this article, we compare the performance of two-valued and m-valued ICs for MOS and bipolar technologies according to speed, chip area, and power dissipation. We present the basic concepts and the basic types of m-valued circuits that make a serious comparison possible.

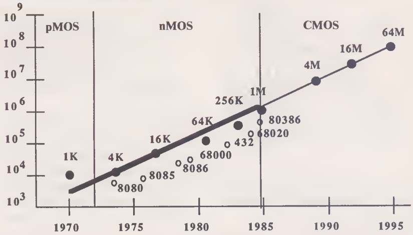


Figure 1. Moore and Noyce law.

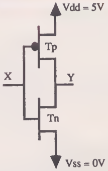


Figure 2. Two-valued CMOS inverter.

Post algebra and m-valued circuitry

Binary integrated circuits implement the basic operators of Boolean algebra. Two logical levels are associated with the 0 and 1 values of Boolean algebra. In the m-valued case, circuits generally implement Post algebra, defined on m different values from 0 to $m-1$ that are totally ordered. Some m-valued circuits don't correspond to Post algebra; current-mode multivalued circuits correspond to the threshold functions, which are a well-known subset of binary functions.

***M*-valued circuits.** Integrated circuits are electrical circuits that use electrical variables to convey information. Generally, the variables are manifested in either voltage or current. In CCD circuits, it is charges. For instance, consider the typical CMOS inverter shown in Figure 2. It uses a 5-volt power supply and consists of an nMOS and a pMOS transistor. When $X < V_{Tn}$, where V_{Tn} is the threshold voltage of the nMOS transistor, T_n is off and T_p is on. The output voltage is $Y = 5\text{V}$. When $X > 5\text{V} - V_{Tp}$ where V_{Tp} is the threshold voltage of the pMOS transistor, T_p is off and T_n is on. The output voltage is $Y = 0\text{V}$. Output Y exhibits two logical states (0V and 5V). But when switching from 0V to 5V and back, Y goes through all the voltages from 0V to 5V. Although only 0V and 5V are used to establish a correspondence with Boolean 0 and 1 elements, the electrical variable (voltage or current) is continuous: $0\text{V} \leq Y \leq 5\text{V}$.

M-valued integrated circuits have more than two discrete logical states. Like two-valued circuits, they use continuous electrical variables to convey information. A correspondence is established between m different voltage or current values and the m different values of the algebra.

We can deduce the following consequence from this fact. As the set of m different voltages is totally ordered, the set of m different values 0 to $m-1$ is totally ordered, as follows: $0 < 1 < 2 < \ldots < m-1$. Thus, the algebra corresponding to m-valued circuits is the Post algebra. When m is a power of 2, a Boolean algebra is possible. The corresponding set of values is partially ordered as shown in Figure 3 for $m = 4$ and $m = 8$. This partial ordering cannot be implemented at the circuit level because voltage (or current) is totally ordered.

Basic operators of Post algebra must be implemented by circuits, even when m is a power of 2. Several sets of operators exist, corresponding to various systems of Post algebra. The monotonic system is the easiest to implement with circuits.[3] To do so, we must define

(1) m different physical levels—voltage levels, current levels, or number of

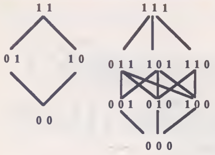


Figure 3. Partial ordering of Boolean algebras.

charges. The circuits with binary inputs that deliver m-valued outputs are called *encoder circuits*; and

(2) circuits to implement $D_i(x)$ and $U_i(x)$ functions of the monotonic system. These circuits, which have m-valued inputs and two-valued outputs are called *decoder circuits*.

The general scheme of m-valued circuits is shown in Figure 4.

The decomposition "decoder—binary synthesis of subfunctions—encoder" results from the use of Post algebra. It exists both in the monotonic representation of Post algebra and at the circuit level.

In some cases, the decomposition is not explicit. For instance, in the case of the m-valued inverter, the two-valued subfunctions are the identity functions and no two-valued circuits are needed. Encoder and decoder circuits can be merged, and the final m-valued inverter is the simplest m-valued circuit that exists. Except for some particular cases, the decomposition exists at the circuit level. The corresponding circuits are level-restoring.

Figure 5 shows the DC ideal characteristic of a level-restoring four-valued inverter. Level-restoring means there are noise margins for each level.

The comparison between m-valued and two-valued circuits deals with

- the circuit complexity of the encoder, decoder, and two-valued circuits that correspond to Figure 4 in the m-valued case; and

Representations of Post algebra

Post algebras are defined on m values for which there exists a totally ordered set.

There are several representations of Post algebra. They have been demonstrated to be isomorphic. For circuit design, it is natural to choose the monotonic system which uses the operators easiest to implement with IC technologies.

The monotonic system has been defined with the following definition and axioms. We present them to show the relationship between Post algebra and its circuit implementation.

Definition

Let m be a fixed integer ≥ 2. A monotonic system of algebra is a distributive lattice M with a zero element 0 and a universal element 1. Operators .(MIN) and +(MAX) are the usual MIN and MAX operators of a lattice. The following axioms are satisfied:

Axiom 1

There exist in M m fixed values $e_0, e_1, \ldots, e_{m-1}$ such that

$$0 = e_0 < e_1 < \ldots < e_{m-1} = 1$$

if $x, e_i \in M$ and $x.e_i = 0$ where $i \neq 0$, then $x = 0$

if $x, e_i, e_j \in M$ and $x + e_i = e_j$ for some $i < j$, then $x = e_j$

Axiom 2

For every $x \in M$, there exists a set of unary operators on x, denoted $D_0(x), D_1(x), \ldots, D_{m-1}(x)$, and $U_0(x), U_1(x), \ldots, U_{m-1}(x)$ where $D_i(x) \in M$ and $U_i(x) \in M$ such that

$D_i(x) = 1$ if $x \leq i$, 0 if $x > i$

$U_i(x) = 0$ if $x < i$, 1 if $x \geq i$

$$x = \sum_{i=1}^{m-1} e_i D_i(x)\, U_i(x)$$

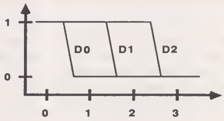
The table gives the D_i and U_i functions when $m = 4$.

From the table, it is obvious that the $D_i(x)$ and $U_i(x)$ functions are implemented by threshold detector circuits. The $D_i(x)$ function is equal to 1 iff $x < i + 0.5$. The $U_i(x)$ function is equal to 1 iff $x > i - 0.5$.

The figure shows graphs of the $D_i(x)$ functions when $m = 4$.

The D_i and U_i functions when $m = 4$.

x	D_0	D_1	D_2	U_1	U_2	U_3
0	1	1	1	0	0	0
1	0	1	1	1	0	0
2	0	0	1	1	1	0
3	0	0	0	1	1	1



Implementation of $D_i(x)$ functions.

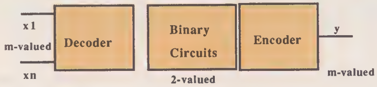


Figure 4. General scheme of *m*-valued circuits.

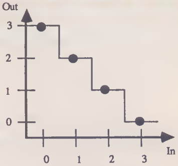


Figure 5. Level-restoring four-valued inverter DC characteristics.

- the circuit complexity of the two-valued circuits that implement the same function in the two-valued case.

This comparison must include propagation delay and chip area.

Threshold functions. A subset of binary functions—the threshold functions—has been considered for a long time. These functions can be expressed as $Y = 1$ if $\Sigma\, a_i x_i \geq T$ for $i = 1$ to n; 0 otherwise. x_i are the input variables, a_i are the weight coefficients, and T is the threshold value. Thus, $a_i, T \in N$.

The specific operations corresponding to threshold functions are analog summation (Σ operator), multiplication by a constant ($a_i x_i$), and comparison with a given threshold value (T). Analog summation is easily implemented by summation of currents (the Kirchoff law of electrical networks). As soon as duplication of current is possible, multiplication of current by a constant is easy; it is realized by adding n times the duplicated current. In binary threshold functions, the input and output are two-valued, and the intermediate value $\Sigma\, a_i x_i$ and the threshold value T are m-valued. The general scheme to implement threshold functions is shown in Figure 6.

Current-mode *m*-valued circuits are suitable for implementing two-valued binary threshold functions. Note, however, that analog summation of current levels is not level-restoring. Noise margins can be critical if levels are not restored between successive stages of logic. Figure 7 shows a typical four-valued analog inverter and the corresponding ideal DC characteristic. This inverter is not level-restoring.

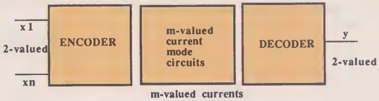


Figure 6. Implementation of threshold functions.

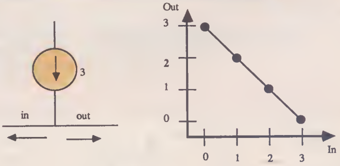


Figure 7. Analog four-valued inverter and DC characteristics.

M-valued voltage mode circuits

In this section, we will examine *m*-valued voltage-mode circuits, focusing on MOS and bipolar technologies because they are the most popular VLSI technologies.

MOS technology. MOS technology first used pMOS transistors and then nMOS transistors. Now, CMOS technology, using both nMOS and pMOS transistors, has replaced nMOS technology as the most popular VLSI technology, mainly because it has no static power dissipation. Basic nMOS technology uses nMOS enhancement-mode transistors and nMOS depletion-mode transistors. Basic CMOS technology uses nMOS enhancement-mode transistors and pMOS enhancement-mode transistors.

For these technologies, each type of transistor has only one threshold voltage, V_T. Using several implants, nMOS technology with several threshold voltages is possible. CMOS technology with depleted-mode n- or p-transistors is also possible. Because these technologies are more expensive, and because yield decreases when process complexity increases, they

nMOS *m*-valued circuits

Here, we show how the threshold and level voltages in the *m*-valued case can be deduced from the binary case.

The figure (at left) shows the two-valued nMOS inverter, where T_L is a depletion-mode transistor, and T_S is an enhancement-mode transistor.

The threshold voltage for T_L is $V_{TL} < 0$ and, for T_S, it is $V_{TS} > 0$. To give the values of the two-valued inverter voltage levels (V_{OH} and V_{OL}) and the switching voltage V_{SW} at which V_O changes from the high level to the low level (see the center figure), we consider the following simplified equations of the MOS transistor:

$$I_D = 0$$

$$\text{if } V_{GS} - V_T < 0$$

$$I_D = K(V_{GS} - V_T)^2$$

$$\text{if } 0 < V_{GS} - V_T < V_{DS}$$

$$I_D = K\, V_{DS}\,[2(V_{GS} - V_T) - V_{DS}]$$

$$\text{if } 0 < V_{DS} < V_{GS} - V_T$$

where V_T is the threshold voltage of the MOS transistor and K is the gain factor:

$K = 0.5\mu_n C_{ox} W/L$ (μ_n is the electron mobility, C_{ox} is the gate capacitance per m^2 and W/L is the width-to-length channel ratio).

According to these equations, the following values of V_{OH}, V_{OL}, and V_{SW} can be easily deduced.

$$V_{OH} = V_{dd}$$

$$V_{OL} = (V_{dd} - V_{TS}) - \sqrt{(V_{dd} - V_{TS})^2 - \frac{K_L}{K_S} V_{TL}^2}$$

$$V_{SW} = V_{TS} - \sqrt{\frac{K_L}{K_S}}\, V_{TL}$$

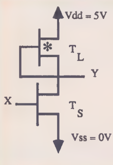
These expressions show that V_{OL} and V_{SW} depend on the threshold voltages V_{TS} and V_{TL} and on the geometry ratio K_L/K_S

$$\frac{K_L}{K_S} = \frac{W_L/L_L}{W_S/L_S}$$

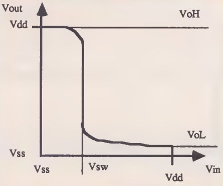
These two possibilities (threshold voltages and geometries of transistors) can be used to realize decoder and encoder circuits of *m*-valued circuits.

From the previous equations, it is obvious that MOS transistors can be used as voltage shifters.

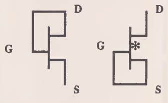
In the figure (at right), with an enhancement-mode transistor, $I_D = K(V_{GS} - V_T)^2$ when $V_{GS} - V_T > 0$ and $V_{DS} = V_T + (I_D/K)^{1/2}$. I_D is fixed by the other transistors of the circuit. V_{DS} can be fixed to a given value according to K (geometry) or to V_T (threshold implant). With a depletion-mode transistor, $I_D = K\, V_{DS}\,[2(-V_{TL}) - V_{DS}]$ when $0 < V_{DS} < -V_{TL}$ and V_{DS} can be fixed to a given value according to K (geometry) or V_{TL} (implants).



Two-valued nMOS Inverter.



nMOS ideal DC transfer curve.



nMOS voltage shifters.

are used only when needed.

For each technology, whether simple or complicated, we will examine how to design the decoder and encoder circuits needed to implement *m*-valued voltage-mode circuits. We will present some examples of *m*-valued circuits and discuss performance.

nMOS *m*-valued circuits. Voltage and threshold levels used in *m*-valued nMOS circuits can be deduced from the binary case. Threshold voltages or geometries of the transistors are the parameters to adjust to obtain the decoder and encoder circuits.

Decoder circuits. These circuits are the threshold detector circuits, with an *m*-valued input and several binary outputs. Using simple nMOS technology, one way to obtain different switching voltages—given that the threshold voltages of enhancement-mode and depletion-mode transistors are fixed—is to define different geometry ratios, as in the two-valued inverter. In such a case, chip area can be important because transistor geometries are fixed by the threshold values V_{TS} and V_{TL} of the technology and the different switching voltages that must be situated between the different output values. Another way to achieve this is to use voltage shifters to shift switching voltages. Figure 8 shows an example of a threshold detector using this technique. T_2 is on only if T_1 is on ($X \geq 2V_T$).

In a technology with several threshold voltages, such as V_{TS}, the first approach can be used with minimum transistor sizes; however, the technology costs more.

Encoder circuits. These have binary inputs and one *m*-valued output. Several options can be used to obtain the *m* different values. The first one uses as many different power supply voltages as needed to obtain $m-1$ values, the final one corresponding to V_{ss}. This option has a major drawback: $m-1$ different power supply lines must be routed through the chip. This solution contradicts the natural evolution of MOS integrated circuits, which began with several power supply voltages (+12V, −12V, +5V), and finally used only +5V voltage.

The second option uses only one power supply voltage, which generally corresponds to the higher voltage level $(m-1)$. The power supply V_{dd} gives the highest level. All the other levels can be generated as the lower level of the two-valued inverter (the set of ratioed transistor sizes, or the set of transistors with different threshold voltages) is obtained.

Figure 9 shows an example of such a four-valued encoder circuit, where *a, b,* and *c* are the binary inputs and *Y* is the four-valued output. In Figure 9, all levels except level 3 are determined by geometry factors K_a, K_b, and K_c. When $a = 1$ and $b = c = 0$, *Y* is determined by the K_l/K_a ratio. When $a = b = c = 1$, *Y* is determined by the $K_l/(K_a + K_b + K_c)$ ratio.

With one power supply, another way to obtain all the levels except the highest and the lowest is through the use of voltage shifts from the low level. Voltage shifts can be realized by ratioed transistor sizes or by transistors with appropriate threshold voltages.

Figure 10 shows an example of a four-valued encoder using this technique. With only one threshold voltage, the voltage shift depends on the transistor geometry (W/L). The other solution is to use a more complicated process, with several voltage thresholds.

These circuits are inefficient for on-chip transmission. Either a more complicated process with several implants is needed, or complicated encoder and decoder circuits using variable geometry transistors must be used, thus leading to important reductions in chip density and speed. With normal technology, *m*-valued circuits are slower and use more chip area.

These circuits are totally unusable for off-chip transmission. As nMOS circuits, they have a static power dissipation that would be prohibitive given the currents needed to drive high capacitive loads. As for nMOS two-valued circuits, some sort of push-pull drivers or CMOS-like circuits are needed for off-chip driving. The only solution is to use several power supply voltages. But this solution is far more complicated than in the two-valued case.

CMOS *m*-valued circuits. Figure 3 presents a CMOS two-valued inverter. Respectively, the output levels are $V_{OH} = V_{dd}$ when T_p is on, and $V_{OL} = V_{ss}$ when T_n is on. Thus, the output levels correspond to the power supply voltages. As with nMOS circuitry, the switching voltage depends on the K_n/K_p ratio, where

$$K_n = \mu_n C_{ox} W_n / L_n$$

$$K_p = \mu_p C_{ox} W_p / L_p$$

$$V_{sw} = \frac{V_{DD} - V_{TP} + \alpha V_{TN}}{1 + \alpha}$$

$$\text{where } \alpha = \sqrt{\frac{K_n}{K_p}}$$

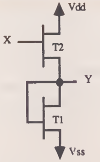
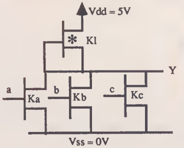
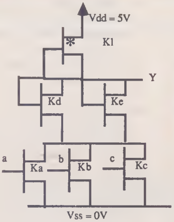


Figure 8. Threshold detection by voltage shift.



a	b	c	Y
0	0	0	3
1	0	0	2
1	1	0	1
1	1	1	0

Figure 9. Example of four-valued nMOS encoder.



a	b	c	Y
0	0	0	3
1	0	0	2
1	1	0	1
1	1	1	0

Figure 10. Four-valued nMOS encoder with voltage shifters.

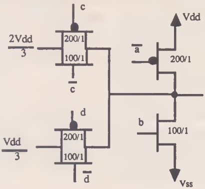


Figure 11. CMOS four-valued encoder circuit.

Encoder circuits. We can obtain the output levels by using switches between the encoder output and the *m* different levels. These levels either correspond to different power rails or are generated by voltage reference circuits. Generally, the switches are the transmission gates, using p and n transistors controlled by complementary inputs, so that both transistors are on or both are off. Figure 11 shows the functional scheme of a four-valued encoder CMOS circuit for pad driving using this technique.[4] Levels 3 and 0 correspond to typical two-valued CMOS. Levels 1 and 2 are realized by CMOS switches corresponding to voltages $2V_{dd}/3$ and $V_{dd}/3$. Two-valued inputs *a, b, c,* and *d* must be chosen properly to ensure that this encoder works properly. Note that all the transistors in Figure 11 are large because the corresponding application is off-chip signaling.

Decoder circuits. As in nMOS circuitry, decoder circuits can be realized with inverters using different sizes of n or p transistors or inverters combined with voltage shifts. Figure 12 shows threshold detectors using inverters plus voltage shifts.[4]

The experimental characteristics of the encoder and decoder circuits for off-chip signaling can be found in Mangin and Current.[4]

Generally, with typical CMOS technology, we need CMOS voltage comparators, with nMOS differential pairs and pMOS loads, to implement the $D_i(x)$ and $U_i(x)$ functions. Some schemes have been presented by Current et al.[5]

Decoder circuits can also use different threshold n or p transistors, or both depletion-mode and enhancement-mode n or p transistors, but the CMOS technology is far more complex for the latter case. We don't give an example because the additional complexity of the process is a serious drawback.

Bipolar technology. For the past 20 years, emitter-coupled logic has been the fastest silicon technology. It has been considered for a long time for *m*-valued circuits, and many circuits have been proposed, either in voltage mode or in current mode.[6] Input signals in ECL circuitry are voltage levels converted into current levels inside the circuit by differential pairs. Currents are then combined linearly and the results are converted back into voltage levels before threshold detection, performed by differential pairs working as voltage comparators. This procedure will be illustrated by circuit examples.

At this point, we should note that the difference between voltage-mode and current-mode ECL circuits corresponds essentially to the application, since circuitry uses both voltage and current. When the application needs *m* different voltage levels for off-chip transmission, *m*-valued circuits are considered voltage mode. When the application corresponds to implementations of threshold functions, the circuits are considered current mode.

In this section, we will present four-valued ECL encoder and decoder circuits[7] designed to reduce the number of wires in an interconnection network. Figure 13 shows the functional scheme for an encoder circuit.

In the figure, the two-valued voltage inputs x and y are converted into a four-valued output current I by summing the currents of the three differential pairs. With three differential pairs instead of two (the minimum), we can obtain a Gray code and simplify the decoder circuit. Four-valued current is converted in a four-valued voltage z by a resistor. Table 1 gives the corresponding truth table.

Figure 14 shows the output decoder functional scheme. The three differential pairs compare the four-valued input z with three threshold levels V_{r1}, V_{r2}, and V_{r3} to get i'_1, i'_2, and i'_3 currents corresponding to $D_0(z)$, $D_1(z)$, and $D_2(z)$ functions on one side of the differential pairs, and i_1, i_2, and i_3 currents corresponding to $U_1(z)$, $U_2(z)$, and $U_3(z)$ functions on the opposite side. Table 2 shows how the currents are combined to obtain the X, CX, Y and CY outputs of Figure 14.

Detailed results were presented in Brilman et al.[7] They show no significant degradation of four-valued switching times versus two-valued switching times. Therefore, single-ended four-valued signaling can be compared fairly to single-ended binary ECL signaling. However, note that differential ECL signaling is preferred in binary cases to single-ended signaling. In four-valued cases, there is no way to use differential signaling. Thus, acceptable switching characteristics are not sufficient arguments for the use of four-valued signaling in practical applications.

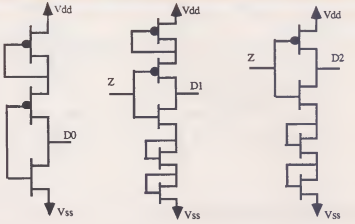
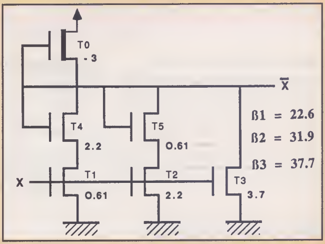


Figure 12. CMOS threshold detectors.

The four-valued nMOS inverter

The four-valued inverter is a particular case of the general scheme of m-valued circuits. Since the binary circuits are not needed, the encoder and decoder circuits can be merged directly to implement the simplest restoring level m-valued circuit that can be realized. The figure shows the schematic circuit of a four-valued nMOS inverter realized with a set of ratioed transistor sizes. In the figure, W/L ratios of transistors T_1 to T_5 are 0.61, 2.2, 3.7, 2.2, and 3.61. β_1, β_2, and β_3 correspond respectively to $(W/L)T_1/(W/L)T_0$, $(W/L)T_2/(W/L)T_0$, and $(W/L)T_3/(W/L)T_0$.

To get β_1, β_2, and β_3 values, one solution is to choose a low value for $(W/L)T_0$; the inverter is very slow. The other solution is to take $(W/L)T_0 = 1$; the inverter area is very large since T_1, T_2, and T_3 areas are important. Using nMOS parameters of the French multiproject chip, we compared the propagation delay through two two-valued inverters and two four-valued inverters, according to the schematic circuit in the figure. The four-valued inverters were about 13 times slower than the binary ones.



Example of nMOS four-valued inverter.

Further reading

Y. Yasuda et al., "Realization of Quaternary Logic Circuits by n-Channel MOS Devices," *IEEE J. Solid-State Circuits*, Feb. 1986, pp. 162-168.

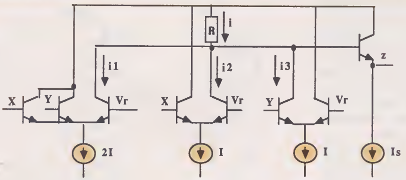


Figure 13. Encoder functional schematic circuit.

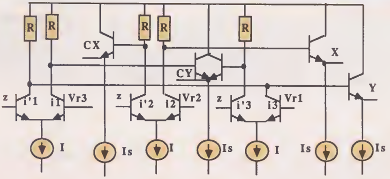


Figure 14. Decoder circuit functional schematic circuit.

Table 1. Encoder circuit truth table.

x	y	i_1	i_2	i_3	I	z
0	0	2	1	0	3	0
0	1	0	1	1	2	1
1	1	0	0	1	1	2
1	0	0	0	0	0	3

Table 2. Decoder circuit truth table.

z	i_1	i'_1	i_2	i'_2	i_3	i'_3	CX	CY	X	Y
0	0	1	0	1	0	1	1	1	0	1
1	1	0	0	1	0	1	1	0	0	1
2	1	0	1	0	0	1	0	0	1	0
3	1	0	1	0	1	0	0	1	1	0

M-valued current mode circuits

In this section, we will examine bipolar circuits and CMOS circuits.

Bipolar technology. Bipolar technology includes integrated injection logic and emitter-coupled logic.

I^2L current mode circuits.[8] Figure 15 shows the basic integrated injection logic gate. In this scheme, the current generator is realized by a pnp bipolar transistor. In that gate, output collectors act as individual sinks to the low voltage level ($V_{ce\,sat}$) when base input current I_b is present. When the input terminal is connected to the low voltage level from a previous gate, base current is diverted away from the multicollector transistor and each output collector becomes a nonconducting path.

I^2L bipolar technology provides an easy implementation of the basic operators needed to implement current-mode *m*-valued circuits. These operators are

- current mirror,
- linear summation of currents,
- thresholding, and
- sink-to-source conversion.

Current-mirror action (see Figure 16) is realized by the feedback connection between one collector and the base, provided the device has sufficient current gain. Current-mirror action realizes current replication and current multiplication by a constant by connecting several collector outputs.

Linear summation of currents corresponds to the Kirchoff law of electrical networks (see Figure 17).

Thresholding involves a current-mirrored transistor and a normal transistor, as shown in Figure 18.

Sink-to-source conversion is realized by a multicollector pnp transistor with current-mirror action, as shown in Figure 19.

We can use I^2L technology to discuss one typical problem of current-mode threshold gates: noise margins. Consider the thresholding example shown in Figure 18. Assuming that the relative tolerances on current values I, threshold values T and current-mirror gains β are respectively δI, δT, and $\delta\beta$, and assuming that $\delta = \delta I = \delta T$, the worst case configurations lead to the final relation

$$\delta \leq \frac{1}{4T} - \frac{\delta\beta_T}{2(1+\beta_T)}$$

where β_T is the current gain of the transistors. In I^2L structure, β_T is low (<10) because the transistor is operating in inverted mode. As an example, if $\beta_T = 6$ with $\delta\beta_T = 0.2$, the maximum relative tolerance on current and threshold values versus the maximum value of T is shown in Table 3. With four-valued currents for which the maximum threshold value is 2.5, tolerance on threshold values and current must be better than 8.5 percent. It should be better than 2.4 percent with eight-valued currents, and better than 0.2 percent with 16-valued currents. With such low β_T current gains, more than four-valued currents cannot reasonably be used.

Tolerance can be improved in I^2L m-valued circuits by increasing the transistor gain β_T. The obvious remedy consists of using an isolated unsaturated high-gain transistor, such as in the integrated Schottky logic structure.

In two-valued circuitry, integrated injection logic has been replaced by the faster and more modern ISL and Schottky transistor logic families. Even if m-valued I^2L circuits have a significant density advantage compared to two-valued I^2L, they're at a considerable speed disadvantage compared to two-valued ISL or STL circuits.

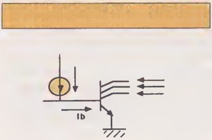


Figure 15. Basic two-valued I^2L gate.

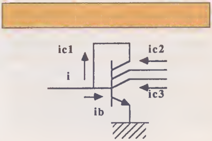


Figure 16. I^2L current mirror.

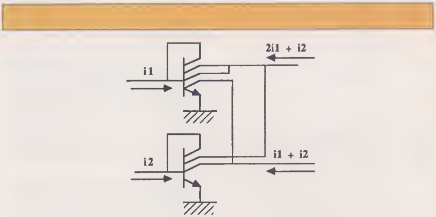


Figure 17. Example of I^2L linear summation.

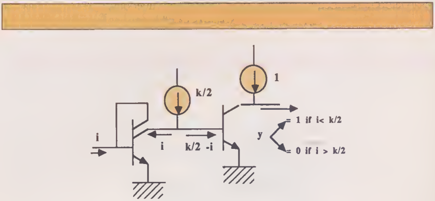


Figure 18. I^2L thresholding example.

ECL current-mode circuits. Emitter-coupled logic m-valued current-mode circuitry has been considered for a long time.

Several versions have been presented by Current.[6] It is interesting to compare them with the multilevel differential logic now used in binary ECL circuitry. Multilevel differential ECL uses several levels of series-gating. Because differential inputs are used, there is no need for voltage references, which are particularly difficult to control in the military temperature range and impossible to control if more than three levels of series-gating are used. With this technique, the only way to realize logic functions is by using series-gating. Any function of *n* inputs can be realized with a complex tree using *n* levels of series-gating. Thus, it has a great logic capability with reasonably low power consumption.

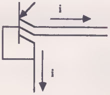


Figure 19. Sink-to-source I^2L conversion.

Table 3. Example of I^2L tolerances.

T	δ
2.5	8.5%
6.5	2.4%
10.5	0.9%
14.5	0.2%

The disadvantage of this technique is that two wires are needed for each variable, with level shifters for any voltage except for the upper level. A detailed comparison with *m*-valued current-mode ECL circuitry has been presented by Etiemble et al.[5] for implementation of threshold functions. Here, we present the results of the comparison.

The usual three-input, two-output adder and the seven-input, three-output adder have been implemented in both versions: multilevel differential circuitry and *m*-valued current-mode circuitry. Comparisons are based on simulation parameters and layout design rules of Thomson 2.5-micrometer advanced bipolar technology.

A layout methodology with simplified design rules has been used for an easy comparison of chip area. The final results of the comparison are given in Table 4. Even if the eight-valued adder is a little slower, it uses one-tenth the binary adder chip area. The four-valued adder is faster and uses less chip area. This comparison shows the advantage of considering *m*-valued current-mode ECL circuits to implement threshold functions.

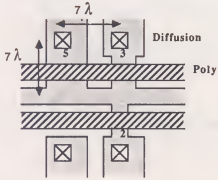


Figure 20. Layout of four ROM cells showing the four transistor sizes.

MOS technology: CMOS current-mode circuits. MOS technology is less efficient than bipolar technology in implementing current-mode circuits because a MOS transistor is voltage-controlled and a bipolar transistor is current-controlled. A current-mode 2-4 encoder and 4-2 decoder were presented by Current et al. in 1987,[5] following presentation of current-mode adders in 1983 and publication of *Proceedings of the International Symposium on Multiple-Valued Logic* in 1985. Current-mode high radix circuits were presented by Onnewheer and Kerkhoff.[5] As with I^2L circuits, the basic elements are constant-current source, current mirror, current comparator, and current switch.

Bidirectional current-mode circuits were presented by Kawahito et al.[5] They use both pMOS current mirrors and nMOS current mirrors, bidirectional current-input circuits, MOS threshold detectors, and voltage-controlled current sources. These circuits have been used to implement radix-4 signed-digit arithmetic. A 32×32-bit multiplier[5] was implemented with modified two-micrometer CMOS technology. Depletion-mode pMOS transistors were added to the usual CMOS process. For a similar propagation delay (59 nanoseconds instead of 56 nanoseconds for the binary multiplier), the multiplier based on *m*-valued bidirectional current-mode circuits dissipates 0.5 watt instead of 1 watt and the chip area is 16.5 square millimeters instead of 30.2 square millimeters. It would be very interesting to obtain information on the tolerances needed for that chip and the corresponding yield.

M-valued current mode CMOS circuits exhibit properties similar to ECL current-mode circuits. With severe constraints on parameter tolerances, they can offer significant improvement in chip area for applications needing threshold functions. The speed is comparable to normal binary CMOS speed.

M-valued memory cells

Memory application is an area where the multivalued approach has been successfully used to design commercial integrated circuits. Read-only memory designs by Intel,[9] Motorola,[10] and General Instrument,[11] and random access memory (RAM) design by Hitachi[12] have been presented. With memories, the basic objective is to reduce chip area while retaining acceptable timing characteristics. ROM and RAM techniques differ.

Read-only memories. The traditional way to achieve high-density ROMs has been to shrink design rule geometries and cell sizes. Standard ROMs store one bit of information in the memory cell as two states. The *m*-valued approach stores more than one bit (generally two) per cell to increase ROM density. Two approaches are possible: the cell uses transistors with variable geometry or with different threshold voltages.

Table 4. Comparison of two-valued and *m*-valued adders.

	3-2 Adder		7-3 Adder	
	Binary	Four-valued	Binary	Eight-valued
tp	1.25 ns	0.95 ns	2.25 ns	4 ns
Area	43,350 λ^2	9,930 λ^2	140,900 λ^2	14,900 λ^2
Pd	18 mW	13.5 mW	47 mW	20 mW

Four-state geometry variable ROM. Intel[9] introduced the four-state geometry variable ROM concept in 1981. The basic idea is to store one out of four states within a single cell, keeping cell size unchanged. Each cell consists of a MOS transistor having one out of the four different impedance values, $Z_0 < Z_1 < Z_2 < Z_3$. Figure 20 shows the layout of four memory cells with four transistor sizes. The different impedances correspond to different transistor channel widths, programmed at the diffusion level. The layout uses only Manhattan design rules (vertical and horizontal lines).

The impedance of a cell transistor is compared to reference transistor impedance $Z_{0.5}$, $Z_{1.5}$, and $Z_{2.5}$ where $Z_i < Z_{i+0.5} < Z_{i+1}$. The basic scheme for reading a memory cell is shown in Figure 21. X_0, X_1, and X_2 outputs deliver the result of comparisons of array transistor impedance with $Z_{0.5}$, $Z_{1.5}$, and $Z_{2.5}$ impedances. Output logic delivers D_0 and D_1 binary outputs corresponding to X_0, X_1, and X_2.

In this approach, cell area is divided by 2. Overhead consists of comparator and decoder circuits. Therefore, the area saving in the Intel nMOS design[9] is 31 percent. The worst-case access time is less than 100 nanoseconds. This approach has been used in the iAPX432 and the 8087 numeric processor.

Motorola presented a CMOS version in 1985. The different impedances correspond to different transistor channel lengths, programmed at the polysilicon level. For the four-state cell, the area savings is 30 percent. A 256-kilobyte ROM has been designed with an n-well HCMOS process using a two-micrometer polysilicon gate length. Typical access time is 200 nanoseconds. This is somewhat slower than a comparable two-state ROM, although not orders-of-magnitude slower.

Four-state multilevel threshold ROM. In this approach, the four possible states stored within a cell are obtained by using multiple ion implants to vary the threshold of the transistor in the cell. By doing so, transistors with four different thresholds can be placed in the cell. Cell area is thus divided by two.

This technique involves a more complicated process, since three masks are needed for programming the ROM instead of the one used in the two-valued case or in the variable geometry approach. The threshold detection of a cell is realized by

I^2L, ISL, STL for binary and *m*-valued circuits

In two-valued circuitry, integrated injection logic has been replaced by integrated Schottky logic and Schottky transistor logic.

Because the I^2L npn bipolar transistor operates in the inverted mode and has no antisaturating feature, it has poor switching characteristics.

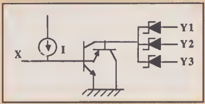
In ISL and STL (see the figures), the npn transistor is operating in direct mode. The pnp transistor has been replaced by a resistor. Output Schottky diodes are used to realize logic functions by wire-ANDing.

In ISL, pnp parasitic transistors are used as antisaturating elements. In STL, a Schottky diode with another built-in voltage antisaturates the npn transistor. The voltage swing is 200 mV instead of about 700 mV in I^2L technology.

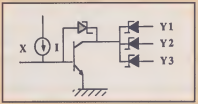
ISL and STL technologies are faster than I^2L. As shown, implementation of *m*-valued I^2L needs both npn and pnp current mirrors. But pnp transistors no longer exist in STL or ISL. The multicollector feature no longer exists. There are no simple *m*-valued versions of ISL and STL technologies.

I^2L *m*-valued circuits have two basic disadvantages compared to ISL and STL technologies, which are the modern versions of I^2L two-valued technology. In I^2L, a tight control of parameter tolerance is needed when *m* increases. In addition, I^2L is far slower than ISL or STL because it has no antisaturating feature and has a higher voltage swing.

Thus, the density advantage of *m*-valued I^2L is not sufficient compared to the speed disadvantage.



ISL inverter.



STL inverter.

linearly ramping the word line; in other words, the input of the detected area transistor. When the word line reaches the threshold level of the area transistor, the transistor turns on.

The type of transistor (one threshold level out of four) is detected by the length of time it takes for it to turn on. Multilevel threshold technique has been used in three devices, including a 128-kilobyte ROM[11] now in full commercial production (through General Instrument). With a six-micrometer nMOS process, the fastest ROM (a 40-kilobyte device) has a typical access time of 500 nanoseconds and a worst-case access time of 800 nanoseconds.

But fast access time was not the objective. Yield has been similar to two-valued ROMs, and density has been increased by close to a factor of two.

Dynamic RAM. The *m*-valued approach is more complicated with RAM since it requires writing in the cells.

No *m*-valued versions of a static RAM have been presented until now. Two-

Comparison of binary and *m*-valued ECL adders

The figure (below, left) shows the implementation of the usual 3-2 adder, which has three binary inputs and two binary outputs (sum and carry). In the figure, the true value of each variable is connected to the left side of differential pairs and to the complementary value on the right side. The level shifters corresponding to x_0 and x_1 inputs are not shown.

As the figure indicates, some differential pairs can be used both for carry and sum outputs. Thus, the multilevel differential pairs implementing carry and sum functions are merged into only one multilevel differential pair to minimize the number of transistors.

The corresponding four-valued 3-2 adder is shown in the figure (below, right). Current inputs are two-valued (0 or I). They are summed in X. The summed current is 0, I, $2I$, or $3I$. The X voltage is subsequently V_{cc}, $V_{cc}-RI$, $V_{cc}-2RI$, and $V_{cc}-3RI$, and the z voltage is $V_{cc}-RI$, $V_{cc}-2RI$, $V_{cc}-3RI$, and $V_{cc}-4RI$ where I is the current of the basic reference current. RI is chosen equal to a voltage diode drop V_D. As series-gating is used, at least V_D voltage shift is needed between the input of the transistor pair referred to as V_{r3} and the input of the transistor pair referred to as V_{r1}. z_{-1} corresponds to the V_D shift from z. References V_{r3} and V_{r1} are $V_{cc}-1.5RI$ and $V_{cc}-3.5RI$, respectively. The operation of the 3-2 four-valued adder is summarized in the table (at right). Sum and carry outputs are two-valued (0 or I).

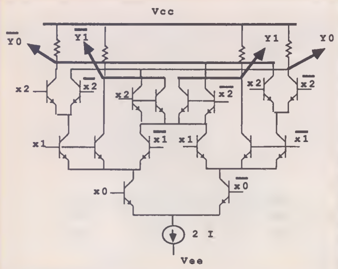
The same comparison has been made between the two different versions of seven-input, three-output adders. In this case, for each version, only one half diode drop ($V_D/2$) is used between differential pair levels to reduce power consumption. So, voltage shifts are realized by a diode (V_D) and a resistor ($RI = V_D/2$).

Four-valued 3 – 2 adder operation.

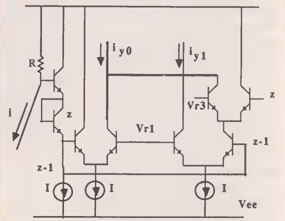
$V_{r3} = V_{cc} - 1.5V_d$

$V_{r1} = V_{cc} - 3.5V_d$

i	z	z_{-1}	i_{y1}	i_{y0}
0	$V_{cc}-V_d$	$V_{cc}-2V_d$	0	0
1	$V_{cc}-2V_d$	$V_{cc}-3V_d$	0	1
2	$V_{cc}-3V_d$	$V_{cc}-4V_d$	I	0
3	$V_{cc}-4V_d$	$V_{cc}-5V_d$	I	1



3-2 binary adder.



M-valued current-mode 3-2 adder.

valued static RAM cells are always a simplified version of a two-valued flip-flop. In the *m*-valued case, the cell is a version of an *m*-valued (*m*-stable) storage element. There are three possible topologies:

(1) true *m*-valued memory elements, with *m*-valued feedback signals;
(2) multiple binary storage circuits, in which bistable circuits with binary feedback are interconnected to give exactly *m* different combinations of stable states; and
(3) a cascade assembly of *m* binary switches in which only one of the elements is in a unique condition for each of the *m*-stable states.

Using any of these methods, the *m*-valued RAM cell area is always greater than several two-valued cells storing the same information.

An *m*-valued version of a dynamic RAM has been presented in Aoki et al.[12] It employs techniques previously used in multilevel CCD memories. Four bits corresponding to 16 different possible charges are stored in the capacitance C_s of a dynamic RAM (DRAM) transistor cell (see Figure 22).

In the write-mode operation, a descending 16-level staircase pulse is applied to the word line; that is, to the selected transistor cell. For the *i*th-level input, the data-line voltage is changed from low to high when the pulse level is *i*. In the read-mode operation, an ascendant staircase pulse is applied to the word line.

The same pulse is transferred to a dummy cell, with the transistor having a $C_s/2$ capacitance for signal comparison. In this technique, data is transferred as a timing signal. No precise timing information is given in Aoki et al.,[12] but it is obvious that cycle time for read or write is long, since it depends on the number of levels of the staircase pulse. If density is multiplied by N, a part of the read- or write-cycle time is multiplied by 2^N; the other part corresponds to the charge transfer preamplifier and the sense amplifier timing characteristics.

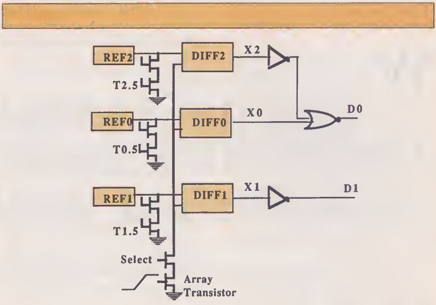


Figure 21. Basic scheme for reading a memory cell.

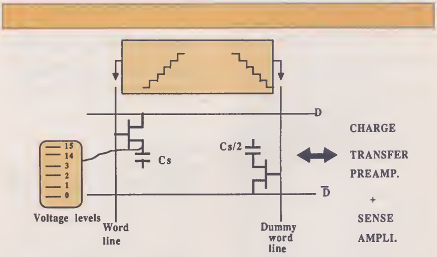


Figure 22. A 16-valued RAM cell.

***M*-valued memories?** The examples of *m*-valued ROMs or RAM that we considered are typical of multivalued ICs. Variable geometry transistor ROM is a good example.

Using French multiproject chip (MPC) λ simplified design rules, we examined the possible transistor sizes for the ROM cell with Manhattan design rules. With nMOS, metal pitch is 7λ, minimum diffusion width is 2λ, and minimum space between diffusion layers is 2λ. According to Figure 20, the transistor width is then $2\lambda < W < 5\lambda$. The four different impedances can be easily obtained with no transistor and transistors with $W = 2\lambda$, $W = 3\lambda$, and $W = 5\lambda$.

With CMOS, metal pitch is 6λ, minimum N^+ diffusion width is 3λ, and minimum distance between N^+ diffusion is 2λ. In this case, transistor width is $3\lambda < W < 4\lambda$.

Therefore, it is impossible to define four different transistor widths without increasing metal pitch. In this case, the gain is less than 50 percent. This approach is less efficient with CMOS than with nMOS. It is technology dependent.

Four-valued nMOS ROM was a good solution and has been used by Intel. That does not mean that it is *the* solution, or the best solution. Four-valued CMOS ROM has been designed by Motorola, but the density advantage versus binary corresponding ROM was not as significant as with nMOS. In addition, access time was not so good.

With 16-valued RAM, the conclusion is

similar. It is a good solution if density is the major objective, but it is a slow device.

We have considered the principal designs of *m*-valued ICs with MOS and bipolar technologies, including voltage-mode, current-mode, and memory circuits.

We have examined the designs of *m*-valued circuits fabricated and used: four-valued ROMs and I^2L circuits implementing threshold functions. Even if they have not been used in practical applications, some other *m*-valued ICs show significant performance, according to some VLSI criteria. Therefore, we conclude that the design of *m*-valued circuits is possible and useful.

At the same time, *m*-valued circuits have not surpassed the two-valued type, since two-valued circuit development is exponential (Figure 1). In this article, we have presented some arguments to explain that paradox. It is related to the nature of *m*-valued circuits; they are generally more complicated than corresponding binary ones implementing the same function. Most of them use Post algebra as a consequence of the totally ordered set of values on voltage or current used to convey information in electrical circuits.

At the circuit level, then, *m*-valued circuits are the transposition of the basic operators of Post algebra. Just as monotonic representation of Post algebra uses *m*-valued to binary conversion ($D_i(x)$ and $U_i(x)$ functions), binary synthesis, and binary to *m*-valued conversion with max, min operators on the *m* values, the *m*-valued circuits use decoder, binary, and encoder circuits.

We have considered the circuit implementation of analog operators, along with the associated tolerance problems. IC design is always a trade-off between many contradictory characteristics: speed, chip area, power dissipation, yield, design cost, and others that comprise the VLSI criteria.

M-valued ICs can present characteristics equivalent and superior to the corresponding two-valued ones according to some VLSI criteria, and worse than two-valued ones according to other criteria. In some applications, some criteria can be decisive. We showed that implementation of threshold functions with ECL *m*-valued circuits results in a significant improvement in chip density with a resultant minor penalty in speed at a given power dissipation.

The importance of each VLSI criteria strongly depends on the application. In some applications, *m*-valued circuits can be a good solution, and even *the* solution. Such an application is the implementation of threshold functions. *M*-valued ROM is another solution. In some very specific applications, pin reduction can be another.

For a long time, researchers in the *m*-valued area have tried to prove the possible advantages of *m*-valued circuits compared to two-valued circuits. *M*-valued circuits and two-valued circuits must not be seen as competitors. If they are seen as such, then two-valued circuits have already won.

The key objective is to carefully examine the domains, the applications where *m*-valued circuits can be useful in the two-valued world. Using this approach, there is no doubt that new *m*-valued circuits will be presented and used in the future. ☐

Acknowledgments

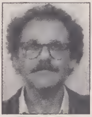
The authors would like to thank Jon T. Butler of the Postgraduate Naval School in Monterey, Calif., and Zvonko Vranesic of the University of Toronto in Ontario, Canada, for helpful comments and corrections on preliminary versions of this article.

References

1. K.C. Smith, "The Prospects of Multivalued Logic: A Technology and Applications View," *IEEE Trans. Computers*, Vol. C-30, 1981, pp. 619-632.
2. S.L. Hurst, "Multiple-Valued Logic—Its Status and its Future," *IEEE Trans. Computers*, Vol. C-33, 1984, pp. 1160-1179.
3. R.Y. Nutter, R.E. Swartwout, and D.C. Rine, "Equivalences and Transformations for Post Multivalued Algebras," *IEEE Trans. Computers*, March 1974, pp. 294-300.
4. J.L. Mangin and W.C. Current, "Characteristics of Prototype CMOS Quaternary Logic Encoder-Decoder Circuits," *IEEE Trans. Computers*, Feb. 1986, pp. 157-161.
5. *Proc. 1987 Int'l Symp. Multiple-Valued Logic*, May 1987, Boston.
6. K.W. Current, "High Density Integrated Computing Circuitry with Multiple-Valued Logic," *IEEE J. Solid-State Circuits*, Feb. 1980, pp. 127-131.
7. M. Brilman et al., "A Four-Valued ECL Encoder and Decoder Circuit," *IEEE J. Solid-State Circuits*, June 1982, pp. 547-552.
8. T.T. Dao, "Threshold I^2L and Its Application to Binary Symmetric Functions and Multivalued Logic," *IEEE J. Solid-State Circuits*, Oct. 1977, pp. 463-472.
9. M. Stark, "Two Bits Per Cell ROM," *Proc. Compcon 1981*, Jan. 1981, pp. 209-212.
10. B. Donoghue, P. Holly, and K. Ilgenstein, "A 256K HCMOS ROM Using a Four-State Cell Approach," *IEEE J. Solid-State Circuits*, Apr. 1985, pp. 598-602.
11. D.A. Rich, K.L.C. Naiff, and K.G. Smalley, "A Four-State ROM Using Multilevel Process Technology," *IEEE J. Solid-State Circuits*, Apr. 1984, pp. 174-179.
12. M. Aoki et al., "A 16 Level/Cell Dynamic Memory," *IEEE J. Solid-State Circuits*, Apr. 1987, pp. 297-299.

Daniel Etiemble is chair of the Computer Science Dept. at the Université P. et M. Curie (Paris 6) in France and a professor of computer science there. His research interests include integrated circuit design and modeling in various silicon and gallium arsenide technologies (with field effect and bipolar transistors), multivalued circuit design, and computer architecture.

Etiemble received an electrical engineering degree from the Institut National des Sciences Appliquées in Lyon in 1969, and both a doctor of engineering degree in computer sciences in 1974 and a doctor of sciences degree in computer sciences in 1979 from Université P. et M. Curie (Paris 6).



Michel Israël is an assistant professor at the Institut d'Informatique d'Entreprise (CNAM), an engineering high school in computer sciences in Evry, France. His research interests include multiple-valued logic circuits, computer-aided design programs for VLSI, and computer architecture design and evaluation. He served as chair of the 12th International Symposium on Multiple-Valued Logic in Paris in 1982 and is currently chair of the Computer Society's Technical Committee on Multiple-Valued Logic.

He received the master of sciences degree in 1973, the third cycle thesis in 1977, and the doctor of sciences degree in 1987, all in computer sciences from the Université P. et M. Curie (Paris 6).

Readers may write to Etiemble, Université P. et M. Curie, Tour 55-65, 4 place Jussieu, 75 252 Paris Cedex 05, France.

A Multiplier Chip with Multiple-Valued Bidirectional Current-Mode Logic Circuits

Michitaka Kameyama, Shoji Kawahito, and Tatsuo Higuchi

Tohoku University

To achieve multiple-valued super chips, we need a multiple-valued hardware algorithm with internal encoding of data. Among the arithmetic applications are many interesting higher-radix number systems suitable for direct multiple-valued encoding, such as the ordinary weighted, the symmetrical weighted, the signed-digit, and the symmetrical residue number systems.

In the signed-digit (SD) number system, the carry propagation during addition and subtraction is always limited to one position to the left, so that totally parallel arithmetic operations can be performed. When we consider implementations of specific arithmetic circuits based on a sign-symmetrical number system such as the SD number system, we can take advantage of multiple-valued bidirectional current-mode circuits. Also, the frequently used linear sum operation can be performed simply by wiring. This property reduces interconnection complexity of the arithmetic large-scale-integration (LSI) chip.

In this article, we describe a 32 × 32-bit SD multiplier implemented with multiple-valued, bidirectional, current-mode circuits and based on two-micrometer complementary metal oxide semiconductor (CMOS) technology.[1]

The multiplier can perform 32-bit two's complement multiplication with three-stage SD full adders using a binary-tree addition scheme. Moreover, the effective multiplier size in the chip and the power dissipation are almost half that of the corresponding binary CMOS multiplier. The multiply time is comparable to that of the fastest binary multiplier. As a result, we have established the effectiveness of the technology for future very large scale integration.

We can achieve super chips using multiple-valued, bidirectional, current-mode logic circuits if we can solve the problems facing implementation.

Advantages of multiple-valued logic circuits

The increasing demand for speed and performance in modern information processing systems clearly points to the need for super chips with tremendous computation power. Low-cost, high-density, fast VLSI devices are essential to make such supercomputing practical in terms of volume, speed, and cost. Almost all digital chips today use conventional binary circuits because of a long history of reliable compact circuitry.

On the other hand, great efforts have also found the advantages of multiple-valued LSI and VLSI implementations over binary ones.[2] Multiple-valued LSI and VLSI have a potential advantage, in that they provide a means of increasing data processing capability per unit area if "multiple-level signals" are used inside the chip.

The advantages of multiple-valued logic circuits are summarized in Figure 1. Multiple-valued logic circuits allow each input/output line to accept and deliver more information, so that the number of pins in LSI and VLSI and the complexity of the interconnections is reduced. If multiple-valued hardware is implemented by partially linear or analog circuits, data processing capability per unit area is increased. For example, multiple-valued current-mode circuits such as integrated injection logic (IIL) and emitter-coupled logic (ECL) consist of linear summers and threshold detectors.[3,4] Analog switches or pass transistors are also useful in multiple-valued circuits used as data selectors.[5] One of the most important advantages of multiple-valued logic can be found at the system level.

However, all the above advantages are not sufficient for complete superiority over binary logic circuits. We must develop a hardware algorithm natural to

0018-9162/88/0400-0043$01.00 © 1988 IEEE

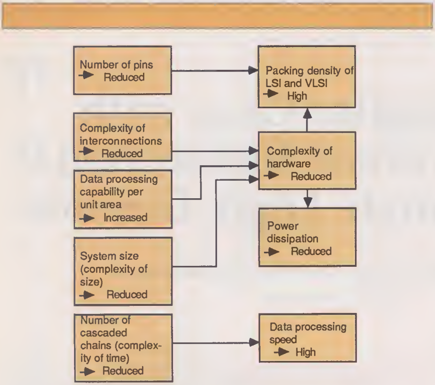

Number of pins
→ Reduced
Packing density of LSI and VLSI
→ High
Complexity of interconnections
→ Reduced
Complexity of hardware
→ Reduced
Data processing capability per unit area
→ Increased
Power dissipation
→ Reduced
System size (complexity of size)
→ Reduced
Number of cascaded chains (complexity of time)
→ Reduced
Data processing speed
→ High

Figure 1. Potential advantages of multiple-valued logic circuits.

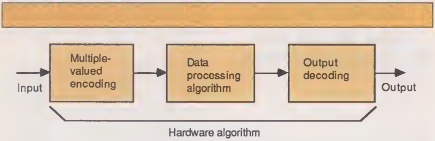


Figure 2. Design strategy of multiple-valued systems.

multiple-valued encoding, as shown in Figure 2. Many different hardware algorithms with encoding schemes can be used to achieve the same input/output specification.

At present, we expect multiple-valued logic to be used in only part of a digital system. Thus, we expect multiple-valued logic circuits to be embedded in systems with binary components, as shown in Figure 3. In this case, code conversions between binary and multiple-valued signals are required. This implies the use of multiple-valued signals with radices that are powers of two, since translation is easiest with this case.

Number systems in arithmetic applications

The choice of a number system is very important. One of the most direct extensions of the ordinary binary number system is the radix r ($r > 2$) weighted number system, defined as

$$A = a_{n-1}r^{n-1} + a_{n-2}r^{n-2} + \dots + a_1 r + a_0$$

where $a_i \in \{0, 1, \dots, r-1\}$, and where A is a non-negative integer number such that $0 \le A \le r^n - 1$. The number of digits necessary to express the dynamic range $0 \le A \le M-1$ is given by

$$n = \lfloor \log_r M \rfloor$$

where $\lfloor x \rfloor$ denotes the smallest integer such that $\lfloor x \rfloor \ge x$. In comparison with the binary system, the total number of digits needed to represent a given number is reduced. For example, a reduction to 63 percent is achieved for the radix 3 and to 50 percent for the radix 4. The properties of the radix-r weighted number system are almost the same as those of the binary number system, so that we can expect an advantage only from the reduced number of digits.

When the radix r is odd and $r = 2k+1$ ($k = 1, 2, \dots$), the balanced representation or the symmetrical number representation becomes useful. In this number system, each digit takes the value $a_i \in \{-k, -(k-1), \dots, 0, \dots, (k-1), k\}$. This sign symmetry renders the arithmetic operation advantageous in that (1) the sign conversion between positive and negative numbers is easily performed by complementing each digit, (2) various signed arithmetic operations can be done without a special conversion technique, and (3) rounding off corresponds to truncation. Several practical circuits have been developed, but their superiority as LSI chips over binary logic systems has not been established.

Avizienis proposed the weighted number system—the signed-digit number representation—in 1961.[6] The differences between the SD and the balanced number systems are that (1) the radix is allowed to be either odd or even in the SD number system, and (2) the SD number system is based on redundant representation, so that the carry propagation chain is limited to a small portion and highly parallel operations can be performed. Let $r = 2k$ ($k = 2, 3, \dots$) for an even radix, and let $r = 2k+1$ ($k = 1, 2, \dots$) for an odd radix. The minimum redundant SD number system with sign symmetry is expressed by the following digit set:

$$a_i \in \{-(k+1), -k, \dots, 0, \dots, k, (k+1)\}$$

Any number A bounded as follows

$$-(k+1)(r^n-1)/(r-1) \le A \le (k+1)(r^n-1)/(r-1)$$

can be expressed by the appropriate choice of each digit. A is not uniquely coded in the SD number system except when $A=0$. For example, the decimal number 10 can be coded as the following four-digit sequences in the radix-4 SD number system:

$$(0\ 0\ 2\ 2),\ (0\ 0\ 3\ -2),\ (0\ 1\ -2\ 2),\ (0\ 1\ -1\ -2),\ (1\ -3\ -2\ 2),\ (1\ -3\ -1\ -2)$$

The detailed SD arithmetic operations are shown in the next section.

Another useful number system is the residue number system (RNS), which is related to 1,700-year-old Chinese arithmetic. The RNS is not a weighted number system. It is defined in terms of a set of relatively prime moduli.[7] The signed RNS is often referred to as the symmetrical system if all the moduli are odd. The symmetrical RNS is constructed from a set of the moduli, $P = \{p_0, p_1, \ldots, p_{N-1}\}$. Let L_i be a symmetrical digit set with respect to the modulus p_i:

$$L_i = \{-(p_i-1)/2, \ldots, 0, \ldots, (p_i-1)/2\}$$

Each residue digit for the number A is defined to be the least absolute value remainder of A when divided by p_i. The RNS holds particular interest because the integer arithmetic operations of addition and multiplication can be performed without the need for carry. The residue digit can be represented by many coding schemes. Multiple-valued coded RNS has been investigated for effective implementation.[7]

As shown in the above number systems, the sign-symmetrical number systems have prominent advantages if the corresponding implementations are carefully considered. In the following sections, we choose the SD arithmetic circuits as examples of the advantageous multiple-valued implementation.

SD hardware algorithm

Addition. We choose radix 4 as the SD arithmetic operation in the implemented LSI because of easy compatibility with binary systems. In the radix-4 SD number system, the addition of two numbers $X = (x_{n-1} \ldots x_i \ldots x_0)$ and $Y = (y_{n-1} \ldots y_i \ldots y_0)$ is performed by the following three successive steps for each digit:

Step 1: $z_i = x_i + y_i$
Step 2: $4c_i + w_i = z_i$
Step 3: $s_i = w_i + c_{i-1}$

where z_i is a linear sum of $x_i \in \{-3, \ldots, 3\}$ and $y_i \in \{-3, \ldots, 3\}$, w_i is an intermediate sum, c_i is a carry, and

$$z_i \in \{-6, \ldots, 0, \ldots, 6\}$$
$$w_i \in \{-2, -1, 0, 1, 2\}$$
$$c_i \in \{-1, 0, 1\} \text{ for } i \ge 0,\ c_{-1} = 0$$

Obviously, $s_i \in \{-3, -2, -1, 0, 1, 2, 3\}$, which is in the same set as the input, so that s_i can be used as a final sum. Therefore, the carry propagation is always limited to one position to the left. This property of the number system allows fast parallel operation, and the addition time is independent of the word length. Subtraction is performed by a similar procedure, complementing the sign of a subtrahend. See Example 1.

Example 1. Let $X = (-3, 2, 3, -1)$ and $Y = (-2, 0, 3, -2)$. Then $S = X + Y$ is obtained as

X:		−3	2	3	−1
+) Y:		−2	0	3	−2
z_i:		−5	2	6	−3
w_i:		−1	2	2	1
c_i:	−1	0	1	−1	
S:	−1	−1	3	1	1

$X = (-3)\times 4^3 + 2\times 4^2 + 3\times 4^1 + (-1)\times 4^0 = -149$
$Y = (-2)\times 4^3 + 3\times 4^1 + (-2)\times 4^0 = -118$
$X + Y = -267$
$S = (-1)\times 4^4 + (-1)\times 4^3 + 3\times 4^2 + 1\times 4^1 + 1\times 4^0 = -267$

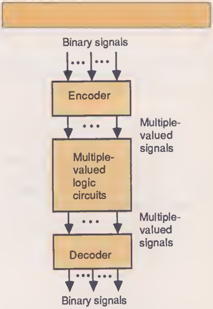


Figure 3. Partial use of multiple-valued logic circuits in binary environments.

Multiplication. The SD number system also has properties useful for multiplication. Parallel multiplication in the radix-4 SD number system consists of two major sections similar to binary multiplication: one generates the partial product and the other adds partial products.

First, let Y be the m-digit multiplier in the radix-4 signed-digit representation, which may be written as

$$Y = \sum_{j=0}^{m-1} y_j 4^j$$

For simplicity, let Y be the coefficient multiplier and let y_j be limited to the value $\in \{-2, -1, 0, 1, 2\}$. With X as the m-digit multiplicand, the product P is given by

$$P = XY = \sum_{j=0}^{m-1} \sum_{i=0}^{m-1} x_i y_j 4^{i+j}$$

where

$$X=\sum_{i=0}^{m-1} x_i 4^i$$

and where $x_i \in \{-3, -2, -1, 0, 1, 2, 3\}$. The partial product z_{ij} as an intermediate result belongs to the following set:

$$z_{ij} = x_i y_j \in \{-6, -4, -3, -2, -1, 0, 1, 2, 3, 4, 6\}$$

Since the dynamic range of the partial product exceeds that of the radix-4 signed digit, we must reduce it by the following two successive steps:

$$4c_{ij} + w_{ij} = z_{ij} \quad \text{for } i = 0, \ldots, m-1$$

$$p_{ij} = w_{ij} + c_{i-1,j} \quad \text{for } i = 0, \ldots, m-1$$

$$p_{mj} = c_{m-1,j}$$

where $j = 0, 1, \ldots, m-1$, and where $c_{ij} \in \{-1, 0, 1\}$ for $i \geq 0$, $c_{-1j} = 0$ and $w_{ij} \in \{-2, -1, 0, 1, 2\}$. Thus, a newly generated partial product becomes $p_{ij} \in \{-3, -2, -1, 0, 1, 2, 3\}$. An operand P_j expressed by $m+1$ signed digits is defined as

$$P_j = (p_{mj} \ldots p_{ij} \ldots p_{0j})$$

The final product P can be obtained by performing the multiple-operand addition of P_j's as follows:

$$P=\sum_{j=0}^{m-1} P_j 4^j$$

The multiplication of operands can be achieved at high speed using two-input parallel SD adders in a binary tree. Figure 4 shows an example of an 8 × 8-digit multiplier constructed as a binary tree. At the first level, each pair of adjacent operands is added together in parallel, producing four intermediate results. Specifically, the number of operands is reduced by half. At the second level, the number of the operands is further reduced by half by the addition of the adjacent two operands. The final product is produced by the addition of the two intermediate results from the

Basic bidirectional current-mode circuits

The bidirectional current-mode circuits are constructed with the following basic circuits, based on CMOS technology.

Current mirror. In the bidirectional current-mode circuits, a current mirror is used for three purposes: (1) to invert the current direction; (2) to produce replicas of an input current; and (3) to scale the input current. There are two types of current mirrors: nMOS and pMOS (see the accompanying figure). Let us consider the principle of the nMOS current mirror as an example. Since the transistor of the input in the current mirror is always operated in the saturation region, the input current I_{in} is given by

$$I_{in} = K(V_{in} - V_T)^2$$

where K, V_T, and V_{in} are, respectively, the proportionality constant, the threshold voltage, and the input voltage. If the transistor of the output operates in the saturation region, the output current of the current mirror is given by

$$I_{out} = K(V_{in} - V_T)^2$$

because the gate is connected to the input of the current mirror. Obviously, I_{out} is equal to I_{in} and independent of the output voltage V_{out}. Therefore, replicas of the input current are produced at the outputs. For short-channel-length devices, however, I_{out} deviates from I_{in} due to the channel length modulation effect.

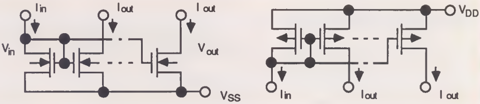
Current source. For fast and stable operation, here we used a current source using a p-channel depletion-mode MOSFET. In the ideal case, the saturation value of the drain current I_d used as a constant current is written as

$$I_d = K_d(W/L)(V_T)^2$$

where K_d, V_T, W, and L are, respectively, the transconductance parameter, the threshold voltage, the channel width, and the channel length of the p-channel depletion-mode MOSFET. The unit current can be set at the specified value using dose control. This type of current source is quite insensitive to the fluctuation of the supply voltage V_{DD}, and requires no bias source or connection other than V_{DD}.

A voltage-switched current source can easily be implemented using a p-channel enhancement-mode MOSFET (current pass transistor) together with the current source (see the accompanying figure).

Threshold detector. The current levels are detected and regenerated by a threshold detector, which can be constructed using the current sources (see the accompanying figure). If the input current I_{in} is smaller than the threshold T, the input voltage of the threshold detector V_{in} is high. This means that the current pass transistor is off and the output current remains 0. On the other hand, if $I_{in} \geq T$, V_{in} is low, the pass transistor turns on and the output current flows.



MOS current mirror:
(a) n-channel type and
(b) p-channel type.

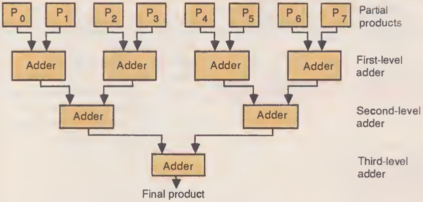


Figure 4. Binary-tree scheme for 8 × 8-digit multiplication.

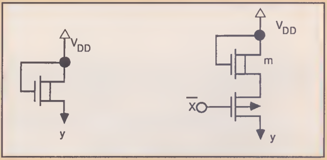
Bidirectional current input circuit. The polarity of the bidirectional current can be detected by a bidirectional current input (BCI) circuit (see the accompanying figure). The relationship between current (*I*) and voltage (*V*) of the BCI is given by

$$I^+ = I_{in}, I^- = 0 \quad \text{if } V_{in} > V_{DD}/2$$

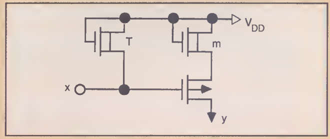
$$I^+ = 0, I^- = 0 \quad \text{if } V_{in} = V_{DD}/2$$

$$I^+ = 0, I^- = I_{in} \quad \text{if } V_{in} < V_{DD}/2$$

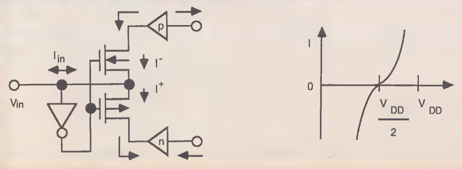
From this equation, it is clear that the direction of the input current can be detected by the input voltage V_{in}. If the input current I_{in} generated by wired summation is positive (when the current flows into the input of BCI), the input voltage V_{in} is pulled up to a value greater than $V_{DD}/2$. Alternatively, if I_{in} is negative (when the current flows out of the input of BCI), V_{in} is pulled down to a value less than $V_{DD}/2$. As a result, if $I_{in} \geq 0$, $I^+ = I_{in}$ and $I^- = 0$, and if $I_{in} < 0$, $I^+ = 0$ and $I^- = I_{in}$. Replicas of I^+ and I^- are obtained at the output of BCI through current mirrors. Consequently, a bidirectional current can be decomposed into two single-directional currents by using BCI.



Current source using a p-channel depletion-mode MOSFET: (a) schematic; (b) voltage-switched current source.



MOS threshold detector.



Bidirectional current input circuit: (a) schematic; (b) input *I-V* characteristic.

second level. As a result, the partial-product addition is completed with three levels of SD adders in the binary-tree scheme. In general, the total number of SD full-adder levels of the binary-tree scheme is given by

$$L = \lfloor \log_2 m \rfloor$$

where m is the number of digits.

Example 2 shows a 4 × 4-digit multiplication for specific choices of the multiplicand and multiplier.

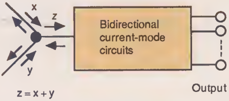


Figure 5. Principle of wired summation in bidirectional current-mode circuits.

Example 2. A 4×4-digit multiplication where the multiplicand X and the multiplier Y are (−2, 3, 3, −2) and (2, −1, 1, −2) are performed as follows:

Multiplier (Y) 2 −1 1 −2							Multiplicand (X) −2	3	3	−2	
(Partial-product generation)							4	−6	−6	4	
						−2	3	3	−2		
$z_{ij} \Rightarrow$					2	−3	−3	2			
				−4	6	6	−4				
						1	−1	−3	−1	0	$\Rightarrow P_0$
$P_{ij} \Rightarrow$						−1	0	−1	−2		$\Rightarrow P_1$
					1	0	1	2			$\Rightarrow P_2$
			−1	1	3	1	0				$\Rightarrow P_3$
Addition First Level:											
$P_0 + P_1 : z_i \Rightarrow$						0	−1	−4	−3	0	
$w_i \Rightarrow$						0	−1	0	1	0	
$c_i \Rightarrow$					0	0	−1	−1	0		
					0	0	−2	−1	1	0	$\Rightarrow S_0$
$P_2 + P_3 : z_i \Rightarrow$			−1	1	4	1	1	2			
$w_i \Rightarrow$			−1	1	0	1	1	2			
$c_i \Rightarrow$		0	0	1	0	0	0				
		0	−1	2	0	1	1	2			$\Rightarrow S_1$
Second Level:											
$S_0 + S_1 : z_i \Rightarrow$		0	−1	2	0	1	−1	1	1	0	
$w_i \Rightarrow$		0	−1	2	0	1	−1	1	1	0	
$c_i \Rightarrow$	0	0	0	0	0	0	0	0	0		
Final Product (P):	0	0	−1	2	0	1	−1	1	1	0	

$X = (-2)\times 4^3 + 3\times 4^2 + 3\times 4^1 + (-2)\times 4^0 = -70$

$Y = 2\times 4^3 + (-1)\times 4^2 + 1\times 4^1 + (-2)\times 4^0 = 114$

$XY = -7980$

$P = (-1)\times 4^7 + 2\times 4^6 + 1\times 4^4 + (-1)\times 4^3 + 1\times 4^2 + 1\times 4^1 = -7980$

Bidirectional, current-mode MOS circuits

The most important principle of multiple-valued current-mode circuits is wired summation as introduced with multiple-valued integrated injection logic.[3] However, "single-directional" current-mode circuits are not always suitable for implementation of arithmetic circuits based on the sign-symmetrical number system. On the other hand, bidirectional current-mode circuits are suitable.[8,9]

Figure 5 illustrates the principle of bidirectional wired summation. From Kirchhoff's current law, the current z equals the sum of the two currents x and y. The current z is applied to successive bidirectional current-mode circuits, where polarity and current levels are detected and arithmetic operations are performed using several basic circuits.

Figure 6 provides the symbols and functions of the available basic bidirectional current-mode circuits. The most basic and important function in the current-mode circuits, threshold detection and regeneration, is implemented by a threshold detector. Current mirrors are also used to produce replicas of an input current.

Single-directional current-mode circuits are also constructed with threshold detectors and current mirrors. For the extension to bidirectional current-mode circuits, we introduce two basic functions: the current mirror inverts the current direction; and bidirectional current input (BCI) circuits convert bidirectional current into single-directional current. (See the sidebar, "Basic bidirectional, current-mode circuits.")

SD adder

Figure 7 shows the parallel SD adder based on bidirectional current-mode circuits. Steps 1 and 3 in the addition of two SD numbers are linear summations, which can be performed by wired summation without active devices. Step 2 can be performed by the signed-digit full adder (SDFA), where we can obtain the intermediate sum w_i and the carry c_i as follows:

$$w_i = z_i - 4,\ c_i = 1 \quad \text{if } z_i \geq 2$$

$$w_i = z_i,\ c_i = 0 \quad \text{if } -1 \leq z_i \leq 1$$

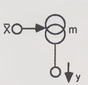
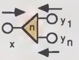
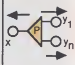
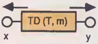
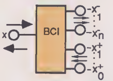
Basic circuit	Current source	Current mirrors		Threshold detector	Bidirectional current input
		nMOS	pMOS		
Symbol					
Function	y=0 if $\overline{X}$="1" y=m if $\overline{X}$="0" $\overline{X}$= input y= output m= integer	$y_i = -a_i x$ for i=1 . . . n x = input y_i = output a_i = scale factor		y=0 if x<T y=m if x≥ T x=input y = output T= threshold m= integer	$x_i^+ = x,\ x_i^- = 0$ if $x \geq 0$ $x_i^+ = 0,\ x_i^- = x$ if $x < 0$ for i = 1 . . . n x = input x_i^+, x_i^- = output

Figure 6. Symbols and functions of basic bidirectional current-mode circuits.

$$w_i = z_i + 4,\ c_i = -1 \quad \text{if } z_i \leq -2$$

Using 26 transistors, we can construct the SDFA from the basic circuits of Figure 6 and the wired summation of Figure 5. To improve speed and DC characteristics, it helps to reduce the number of cascaded current mirrors in the SDFA. One solution is to retain the intermediate sum w_i in inverted form. This is shown in Figure 8, which represents the SDFA block of Figure 7.

In general, signal levels must be restored before their complete degradation. While the carry c_i is quantized by the threshold detector, the intermediate sum w_i is not quantized in Figure 8. Thus, an inverted quantizer is essential for the output w_i. At the same time, the specified polarity can be

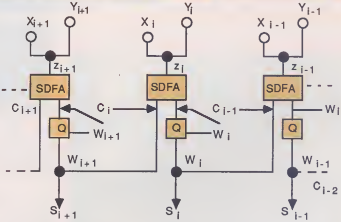


Figure 7. Parallel signed-digit adder.

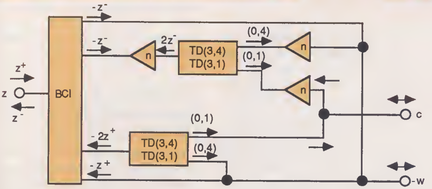
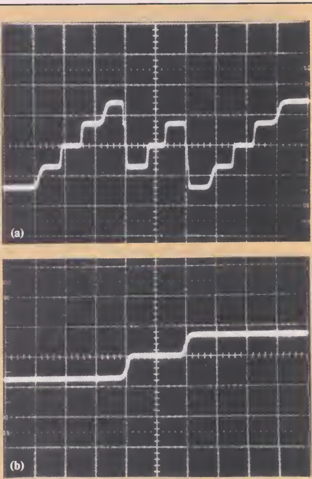


Figure 8. Signed-digit full adder.

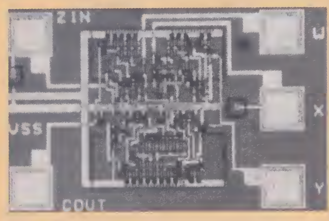
Characteristics of the SD full adder

A test chip of the signed-digit full adder with an inverted quantizer has been implemented using two-micrometer CMOS technology (see the accompanying figure). The effective size is about 140×205 square micrometers, which is almost the same as for two-bit binary CMOS full adders. The current transfer curves of the implemented circuits (see the accompanying two-part figure) show the unit current to be approximately 31 microamperes with transistor sizes of $L = 2.8$ micrometers and $W = 9$ micrometers, and a threshold voltage of 1.5 volts. The accompanying table clearly shows the successful operation of the implemented circuit.

The minimum noise margin is about 10 microamperes (30 percent of the unit current). Let us consider the noise margin with respect to a device parameter deviation. If the statistical variation from the desired output of the current source exceeds the noise margin, logical errors will occur. The variation of the current source output current ΔI is mainly caused by the variation of the threshold voltage ΔV_T of the transistor. The variation is represented as $\Delta I = (\Delta V_T / V_T)(2 + \Delta V_T / V_T) I_a$. In the case where $\Delta V_T = 50$ millivolts, $V_T = 1.5$ volts, and $I_a = 31$ microamperes, ΔI becomes about two microamperes. This variation is not large enough to cause logical errors.



DC transfer characteristics of the SD full adder with inverted quantizer: (a) intermediate-sum output; (b) carry output. (Units of measurement are 41 microamperes per division on the horizontal and vertical axes.)



Implemented SD full adder with inverted quantizer.

Input z	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
Intermediate sum :w	-2	-1	0	1	2	-1	0	1	-2	-1	0	1	2
Carry :c	-1	-1	-1	-1	-1	0	0	0	1	1	1	1	1

Relation between the input and output of SDFA.

obtained through the inversion operation.

For a detailed explanation of the SDFA, see the sidebar "Characteristics of the SD full adder."

The SD adder constructed with bidirectional current-mode circuits has a very simple structure. Clearly, the addition speed is independent of the word length, determined only by the delays of the intermediate sum output of SDFA and the inverted quantizer.

Figure 9a shows the addition time of the parallel SD adder compared to the binary adders. The SD adder is much faster than conventional binary adders, especially when the word length is very long.

Figure 9b shows the number of transistors required in the CMOS n-bit adders. The SD adder requires fewer transistors than the binary adders. Also, the interconnections among the modules are simple and fewer in number when compared to conventional high-speed adders such as a carry lookahead adder (CLA), a carry select adder (CSA), and so on. Consequently, the SD adder is superior to the conventional binary adder in both speed and compactness.

LSI design and implementation of the SD multiplier

The multiplication algorithm presented before is also useful for signed-binary number (two's complement) inputs. We implemented a 32×32-bit multiple-valued SD multiplier with binary inputs, based on

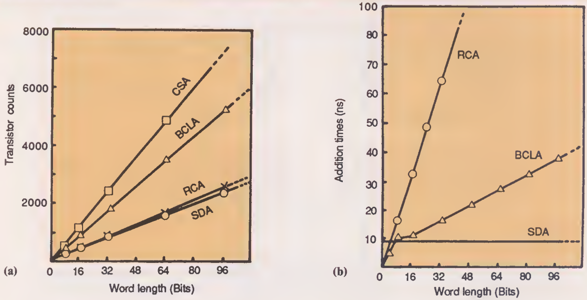


Figure 9. Comparison of various parallel adders: (a) transistor count and (b) addi tion time. (SDA: signed-digit adder; RCA: ripple-carry adder; BCLA: block carry- lookahead adder; CSA: carry-select adder.)

two-micrometer CMOS technology. Figure 10 shows the block diagram of the multiplier.

One of the most important techniques for the realization of a high-speed compact multiplier is to reduce the number of partial products. To reduce the number of partial products, binary multipliers often use the modified Booth's algorithm.[10] By an extension of the modified Booth's algorithm, we can use a recoding algorithm suitable for radix-4 SD number partial-product generation.

In our system, the multiplier coefficient *Y* is divided into eight groups of five bits each. Each pair of two contiguous groups has one bit in common. First, from the multiplicand *X* and the recoded signals, the partial products are generated in the partial-product generator (PPG) by binary logic circuits such as multiplexers and exclusive OR gates. At the output of PPG, the partial products are converted into radix-4 signed digits represented by current-mode signals. Consequently, eight operands composed of 18-digit partial products are produced. (For further discussion, see the sidebar "Recoding algorithm for radix-4 SD number multiplication.")

A binary-tree structure is also used for the addition of the partial products. Figure 11 shows the binary-tree addition scheme using bidirectional current-mode SD adders. Obviously, the 32 × 32-bit mul-

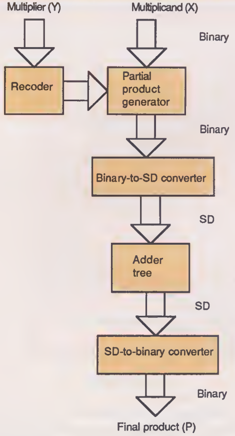


Figure 10. Block diagram of the signed-digit multiplier with binary inputs.

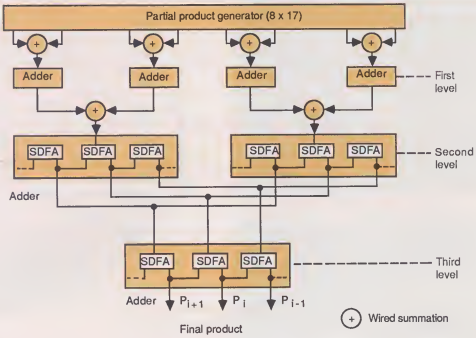


Figure 11. Binary-tree scheme of the 32 × 32-bit multiplier.

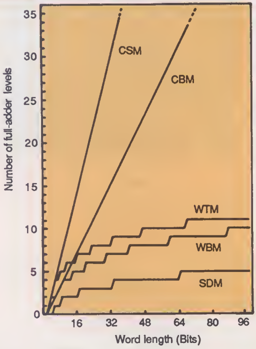


Figure 12. Comparison of the number of full-adder levels in various multipliers. (SDM: proposed SD multiplier; WBM: binary multiplier with Wallace Tree (WT) and modified Booth's algorithm (MBA); WTM: binary multiplier with WT; CBM: binary multiplier with carry-save adder (CSA) and MBA; CSM: binary multiplier with CSA.)

tiplication can be performed by the delays of three levels of full adders. The bidirectional wired summations are fully used, greatly reducing the number of active devices and the complexity of the interconnections. We found the binary-tree structure very useful for the LSI implementation because of its regularity.

Generally, the number of full-adder levels L in the proposed $n \times n$-bit multiplier is given by

$$L = \lfloor \log_2(n/4) \rfloor = \lfloor \log_2 n - 2 \rfloor$$

because the number of operands composed of $(n+2)$-digit partial products are reduced to $n/4$ by the recoding algorithm.

Figure 12 shows the relation between the number of full-adder levels and the word length for various multipliers. The number of full-adder levels in a binary multiplier is defined as the number of full-adder chains necessary to reduce all the partial products to two operands. In this case, we usually need a carry-lookahead adder for the final product. The number of full-adder levels for the SD multiplier (SDM) is less than half the number of the fastest binary multiplier (WBM)

Recoding algorithm for radix-4 SD number multiplication

The recoding algorithm is effective for the reduction of the number of partial products, which leads to the reduction of the number of the cascaded SD full-adder chains and the number of the SD full adders.

Let Y be the multiplier in an n-bit two's complement number. Assume that n is a multiple of four. The multiplier Y may be written as

$$Y = -y_{n-1}2^{n-1} + \sum_{i=0}^{n-2} y_i 2^i$$

where $y_i \in \{0, 1\}$, which can also be expressed as follows:

$$\begin{aligned} Y &= (y_{n-5} + y_{n-4} + 2y_{n-3} + 4y_{n-2} - 8y_{n-1})2^{(n-4)} \\ &\quad + \qquad \cdots \\ &\quad + (y_{4j-1} + y_{4j} + 2y_{4j+1} + 4y_{4j+2} - 8y_{4j+3})2^{4j} \\ &\quad + \qquad \cdots \\ &\quad + (y_3 + y_4 + 2y_5 + 4y_6 - 8y_7)2^4 \\ &\quad + (y_{-1} + y_0 + 2y_1 + 4y_2 - 8y_3)2^0 \\ &= \sum_{j=0}^{n/4-1} (y_{4j-1} + y_{4j} + 2y_{4j+1} + 4y_{4j+2} - 8y_{4j+3})2^{4j} \\ &= \sum_{j=0}^{n/4-1} Q_j 16^j \end{aligned}$$

where $Q_j \in \{-8, \ldots, 0, \ldots, 8\}$ is defined as

$$Q_j = (y_{4j-1} + y_{4j} + 2y_{4j+1} + 4y_{4j+2} - 8y_{4j+3})$$

and $y_{-1} = 0$. Consequently, the multiplier Y can be divided into $(n/4)$ groups of five bits each. Each pair of two contiguous groups has one bit in common. With X the multiplicand, the product P is represented as

$$P = \sum_{j=0}^{(n/4)-1} Q_j X 16^j$$

The basic concept of the algorithm is to generate the product Q_jX by shifting and complementing X. The value Q_j can be decomposed into appropriately selected values U_j and V_j as

$$Q_j = U_j + V_j$$

where $U_j \in \{-2, -1, 0, 1, 2\}$ and $V_j \in \{-8, -4, 0, 4, 8\}$.

The product Q_jX can be written as

$$\begin{aligned} Q_jX &= U_jX + V_jX \\ &= A_j + B_j \end{aligned}$$

If A_j and B_j are expanded into two's complement numbers, they become

$$A_j = U_jX = -a_{nj} + \sum_{i=0}^{n-1} a_{ij}2^i$$

$$B_j = V_jX = -b_{(n+2)j} + \sum_{i=0}^{n+1} b_{ij}2^i$$

Since the multiplicand X is represented as a two's complement binary number, we can generate A_j and B_j by shift and complement operations. For example, the values $8X$, $4X$, and $2X$ correspond to three-bit, two-bit and one-bit left-shift operations in X, respectively. The value $-X$ can be obtained by first complementing the number X, then adding one to its least significant bit.

To generate the partial product represented by the radix-4 SD number, consider a weighted linear sum z_{ij} defined as

$$Z_j = \sum_{i=0}^{n/2} z_{ij}4^i = Q_jX$$

The weighted linear sum z_{ij} can be generated from each of two bits in A_j and B_j as follows:

$$z_{ij} = 2b_{(2i+1)j} + b_{2ij} + 2a_{(2i+1)j} + a_{2ij}$$

$$\text{for } i \neq n/2$$

$$z_{(n/2)j} = -4b_{(n+2)j} + 2b_{(n+1)j} + b_{nj} - a_{nj}$$

where $z_{ij} \in \{0, \ldots, 6\}$ for $i \neq n/2$, and $z_{(n/2)j} \in \{-5, \ldots, 3\}$. Since $-5 \leq z_{ij} \leq 6$, z_{ij} can be transformed into the SD number representation

$$P_j = \sum_{i=0}^{(n/2)+1} p_{ij}4^i$$

by the following two steps:

$$4c_{ij} + w_{ij} = z_{ij}$$

$$p_{ij} = w_{ij} + c_{(i-1)j}$$

$$\text{for } i \neq (n/2)+1$$

$$p_{[(n/2)+1]j} = c_{(n/2)j}$$

where $w_{ij} \in \{-2, -1, 0, 1, 2\}$, $c_{ij} \in \{-1, 0, 1\}$, and $p_{ij} \in \{-3, -2, -1, 0, 1, 2, 3\}$.

Example. Let $Y = (1010)_2$ and $X = (1011)_2$ in the two's complement form. Then the final product P is obtained as follows:

$$Q_0 = -8 \times 1 + 2 \times 1 = -6,\ U_0 = 2,\ V_0 = -8$$

$$\begin{array}{lrrrrrrr} & & & -a_{40} & a_{30} & a_{20} & a_{10} & a_{00} \\ U_0 = 2 \ \text{:1-bit shift} & & & -(x_3) & (x_2) & (x_1) & (x_0) & (x_{-1}) \\ & & & -(1) & (0) & (1) & (1) & (0) \\ V_0 = -8\text{: 3-bit shift} & -b_{60} & b_{50} & b_{40} & b_{30} & b_{20} & b_{10} & b_{00} \\ & -(\bar{x}_3) & (\bar{x}_2) & (\bar{x}_1) & (\bar{x}_0) & (\bar{x}_{-1}) & (\bar{x}_{-2}) & (\bar{x}_{-3}) \\ \text{complement} & -(0) & (1) & (0) & (0) & (1) & (1) & (1) \\ & & & z_{20}=1 & & z_{10}=2 & & z_{00}=5 \\ w_{i0}: & & & 1 & & 2 & & 1 \\ c_{i0}: & 0 & & 0 & & 1 & & \\ \text{increment} & & & & & & & +1 \\ \hline P: & 0 & & 1 & & 3 & & 2 \end{array}$$

$$\begin{aligned} X &= -1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 = -5 \\ Y &= -1 \times 2^3 + 1 \times 2^1 \qquad\quad = -6 \\ XY &= 30 \\ P &= 1 \times 4^2 + 3 \times 4^1 + 2 \times 4^0 = 30 \end{aligned}$$

using a Wallace tree and modified Booth's algorithm for a word length of 32 bits. Also, we can achieve a 10-fold reduction of the number of adder delays compared to a 32 × 32-bit ordinary array multiplier (CSM).

The final stage of the multiplier is an SD-to-binary converter where the 32-digit SD number is converted into the 64-bit two's complement binary number. We introduce a high-speed conversion technique by using a radix-4 carry lookahead adder.

Figure 13 shows a photomicrograph of the multiple-valued multiplier. Clearly, the structure is very regular, as in memory. Also, the interconnections between the modules are very simple. The chip measures 7.00 × 4.92 square millimeters. The effective multiplier size is 5.23 × 3.16 square millimeters, which is half that of the fastest conventional binary multiplier.

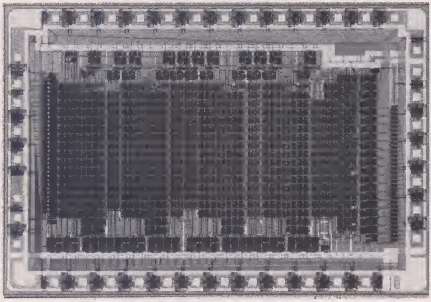


Figure 13. Photomicrograph of the multiplier chip.

Table 1. Comparison of two implemented 32 × 32-bit multipliers.

	Multiple-Valued Multiplier	Fastest Binary Multiplier
Multiply Time (ns)	59	56
Number of Interconnections	200	1,500
Number of Transistors (current mode)	23,600 (7,200)	45,000
Effective Multiplier Size (mm^2)	5.2 × 3.2 (16.6)	5.3 × 5.7 (30.2)
Power dissipation (W)	0.5	1
Technology	2 μm CMOS and depletion-mode pMOS	2μm CMOS

Table 1 compares the implemented SD multiplier and the fastest binary multiplier presented by Gamal.[11] The multiply time is comparable to that of the fastest binary multiplier, because the SD multiplier has half the number of cascaded adder chains and because the delay time of the SDFA module is about twice that of a binary full adder.

The number of transistors is 23,600, which is approximately 52 percent of those needed in a binary multiplier. However, the active area is determined not only by the number of active devices but also by the sizes of the active devices and the number of interconnections. The transistors used for the current-mode circuits are relatively large compared to those used for CMOS gates, while the number of interconnections among the modules is much smaller than in the fastest binary multiplier using a Wallace tree. Also, the regularity of the layout greatly contributes to the reduction of the interconnection area in the multiple-valued multiplier. As a result, a 46-percent reduction of the active area is achieved compared to the binary multiplier.

The multiple-valued multiplier is superior to the conventional multiplier with regard to power dissipation because of the reduction of active devices. The number of transistors used for current-mode circuits is about 7,200; the other transistors are used for CMOS gates.

Applications and future trends

The advantages of signed-digit arithmetic circuits based on multiple-valued encoding have been demonstrated by the LSI implementation of the multiplier. With application to binary digital systems, however, the advantage of fast parallel processing in SD arithmetic may be lost by the conversion between the SD and the binary number systems. If the conversions are performed only at the input and the output of the system and the SD number representation is fully used inside the system, the disadvantage of the conversion can be greatly reduced.

Figure 14 shows an example of a finite impulse response (FIR) digital filter using multiple-valued SD arithmetic circuits. In the coefficient multipliers, the SD-to-binary converter is removed so that the addition of their outputs can be performed by the multiple-valued SD adder. The SD-to-binary conversion is performed at the final stage of the system.

We expect the processing time to greatly improve upon that of the system using conventional binary arithmetic. A commercial digital signal processor based on this concept and using the multiple-valued SD adder and the multiplier is also planned for the near future.

The VLSI implementation of the proposed multiple-valued arithmetic circuits using submicrometer technology still involves a few problems. For example, a decreased noise margin results from the channel-length modulation effect of the device used for the current source and current mirrors. One solution for this problem is to use a special device with the reduced channel-length modulation effect, such as a diffusion self-aligned MOS device.[12] If these problems are successfully solved, we may well see a super chip using multiple-valued logic circuits, such as a subnanosecond 32 × 32-bit multiplier.☐

Acknowledgments

We wish to thank H. Yamada, S. Horiuchi, and T. Ishihara of Matsushita Electric Industrial Co. for many helpful comments. We would also like to thank J.T. Butler of the Naval Postgraduate School for many useful suggestions.

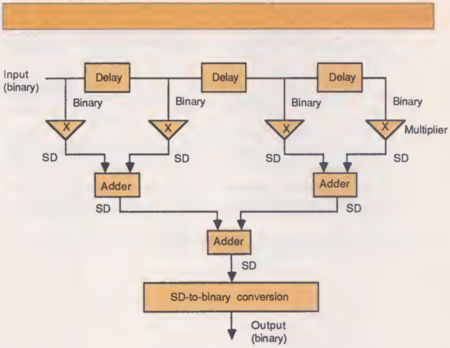


Figure 14. Application to a high-performance finite impulse response digital filter.

Have you heard about our...

Technical Committee on Multiple-Valued Logic?

For information on this, or any of our 32 other TCs, members* may contact

THE COMPUTER SOCIETY
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
Telephone: (714) 821-8380

***Nonmembers:** Join us. Use the application on page 115, or for more information, circle number 202 on the reader service card.

Attention Engineers!

NEED HELP WITH SCREENS, DEMOS AND GRAPHICS?

Try the **DisplayExpress**™ and **C-Display Librarian**™! This software can improve your competitive position and increase your productivity. SATISFACTION GUARANTEED OR YOUR MONEY BACK.

"Tremendous productivity tools"
*(**PC Week** October 27, 1987)*

"Many applications will benefit from the quick development times and minimal programming required to implement a finished, quality product"
*(**IEEE Software** November, 1987)*

"If you want to include bit-mapped graphics in your application, then this might be the package for you"
*(**Computer Language** June, 1987)*

DisplayExpress™ $69
C-Display Librarian™ $69

Send check or money order now to:

SYDETECH
SYSTEM DEVELOPMENT TECHNOLOGIES, INC.
43-23 COLDEN STREET, #17C, FLUSHING, N.Y. 11355

For COD orders or the name of the nearest dealer please call (718) 886-0221

Reader Service Number 3

References

1. S. Kawahito et al., "A High-Speed Compact Multiplier Based on Multiple-Valued Bi-Directional Current-Mode Circuits," *Proc. 1987 Int'l Symp. Multiple-Valued Logic*, May 1987, pp. 172-180.
2. S.L. Hurst,"Multiple-Valued Logic—Its Status and Its Future," *IEEE Trans. Computers*, Dec. 1984, pp. 1160-1179.
3. T.T. Dao, E.J. McCluskey, and L.K. Russell, "Multivalued Integrated Injection Logic," *IEEE Trans. Computers*, Dec. 1977, pp. 1233-1241.
4. K.W. Current, "High Density Integrated Computing Circuitry with Multiple-Valued Logic," *IEEE J. Solid-State Circuits*, Feb. 1980, pp. 127-131.
5. M. Kameyama, T. Hanyu, and T.Higuchi, "Design and Implementation of Quaternary NMOS Integrated Circuits for Pipelined Image Processing," *IEEE J. Solid-State Circuits*, Feb. 1987, pp. 20-27.
6. A. Avizienis, "Signed-Digit Number Representations for Fast Parallel Arithmetic," *IRE Trans. Elect. Computing*, Sept. 1961, pp. 389-400.
7. M.A. Soderstrand et al., eds., *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*, IEEE Press, New York, 1986.
8. L.P. Rubinfield, "A Proof of the Modified Booth's Algorithm for Multiplication," *IEEE Trans. Computers*, Oct. 1975, pp. 1014-1015.
9. S. Kawahito, M. Kameyama, and T. Higuchi, "VLSI-Oriented Bi-Directional Current-Mode Arithmetic Circuits Based on the Radix-4 Signed-Digit Number System," *Proc. 1986 Int'l Symp. Multiple-Valued Logic*, May 1986, pp. 70-77.
10. M. Kameyama and T. Higuchi, "Design of Radix-4 Signed-Digit Arithmetic Circuits for Digital Filtering," *Proc. 1980 Int'l Symp. Multiple-Valued Logic*, June 1980, pp. 272-277.
11. A.E. Gamal et al., "A CMOS 32b Wallace Tree Multiplier-Accumulator," *Dig. IEEE Int'l Solid-State Circuit Conf.*, THPM 15.5, 1986.
12. T. Tarui et al., "Diffusion Self-Aligned Enhance-Depletion MOS-IC (DSA-MOS-IC)," *Proc. 2nd Conf. on Solid-State Devices*, 1970, p. 193.

Michitaka Kameyama is an associate professor in the department of electronic engineering at Tohoku University, Japan. His research interests include multiple-valued logic systems, VLSI-oriented special-purpose processors, highly reliable digital systems, and robotics.

Kameyama was the program co-chair of the 1986 IEEE International Symposium on Multiple-Valued Logic. He received the Award for Excellence at the 1984 and 1985 IEEE International Symposiums for Multiple-Valued Logic and the Technical Excellence Award from the Society of Instrument and Control Engineers of Japan in 1986.

He received the BE, ME, and DE degrees in electronic engineering from Tohoku University in 1973, 1975, and 1978, respectively.

Shoji Kawahito is a research associate with the department of electronic engineering at Tohoku University, Japan. His research interests include multiple-valued logic systems, high-performance arithmetic VLSI, and special-purpose integrated circuits for signal processing.

Kawahito received the BE and ME degrees in electrical and electronic engineering from Toyohashi University of Technology in 1983 and 1985, respectively. He received the DE degree in electronic engineering from Tohoku University in 1988.

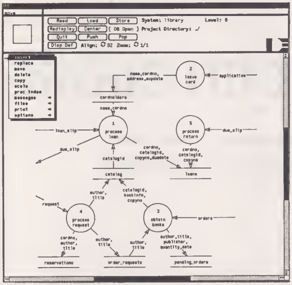
Tatsuo Higuchi is a professor in the department of electronic engineering at Tohoku University, Japan. His research interests include digital signal processing, multiple-valued logic systems, and computation in VLSI.

Higuchi was the program chair of the 1983 IEEE International Symposium on Multiple-Valued Logic. He received the Award for Excellence at the 1984 and 1985 IEEE International Symposiums on Multiple-Valued Logic. He received the Outstanding Transactions Paper award from the Society of Instrument and Control Engineers of Japan in 1984 and 1986.

He received the BE, ME, and DE degrees in electronic engineering from Tohoku University in 1962, 1964, and 1969, respectively.

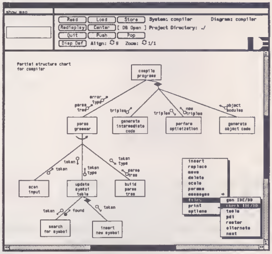
Readers may write to Kameyama at the Dept. of Electrical Engineering, Tohoku Univeristy, Sendai 980, Japan.

IDE is the state-of-the-art in computer-aided software engineering



Software through Pictures™
graphical editors

- Dataflow Diagram Editor
- Structure Chart Editor
- Data Structure Editor
- Entity Relationship Editor
- Transition Diagram Editor



Interactive Development Environments offers a powerful family of graphical editors and an integrated software development environment.

IDE's workstation-based tool environment includes:

- support for several widely used analysis and design methods, including Structured Systems Analysis, Structured Design, and User Software Engineering, with *extensive checking* built into the tools
- the IDE Data Dictionary system, which can be edited directly from the graphical editors; it supports definition of names, along with types, constraints, aliases, and associated text
- rapid prototyping and application generation with the RAPID/USE system
- support for real-time extensions to structured analysis
- Visible Connections™, an open software architecture to support extensions to the tools
- a compact and efficient relational database management system to support the project database and IDE Data Dictionary
- consistent user interfaces to the graphical tools
- a powerful tools library to support additional tool and application development
- available now for the Sun, Apollo, VAXstation and Hewlett-Packard Workstations

Interactive Development Environments is pushing back the frontiers of technology in computer-aided software engineering. Our tools represent an *outstanding* value for their performance and capabilities, with substantial discounts available to universities for teaching and research. Contact IDE and see how we can help to improve your software engineering environment. Call us, telex us, or send us electronic mail at ucbvax!sun!ide!sales.

INTERACTIVE
DEVELOPMENT
ENVIRONMENTS

Sales and Information:
Tel: (415) 543-0900

150 Fourth Street, Suite 210 • San Francisco, CA 94103 • Telex: 256 697 IDE UR

™Software through Pictures and Visible Connections are trademarks of Interactive Development Environments, Inc.
VAX station is a trademark of Digital Equipment Corporation.

Reader Service Number 4

Multiple-Valued CCD Circuits

Jon T. Butler, Naval Postgraduate School, United States
Hans G. Kerkhoff, University of Twente, The Netherlands

The foundation of any computing system is the logic on which it is built. Indeed, the impact of computing on modern technology largely results from the success integrated circuit manufacturers have had in implementing complex circuits compactly. During the 1970s, circuit density doubled every year. However, limits in circuit reduction are now being felt, and the pace has slowed to a doubling every two years, with further slowing expected. This has inspired an interest in innovative ways to achieve still smaller circuits.

Currently, important technologies include emitter coupled logic (ECL), n-type metal oxide semiconductor (nMOS), and complementary metal oxide semiconductor. CMOS in particular predominates because of its low power dissipation, high packing density, and reliability. CMOS devices produce very little heat, making the design of compact systems possible. Power is dissipated only when logic values change; the actual storage of a logic value dissipates almost no power.

Unlike CMOS, which uses voltage to store logic values, charge-coupled device (CCD) technology uses charge. Because CCD has many fewer source-drain connections, extremely compact circuits are possible. Also unlike CMOS, multiple values are easily stored, resulting in even more compact circuits. For example, in prototype four-valued CCD designs,[1] logic values are measured by the number of electrons. In such designs, a logic 0 corresponds to zero electrons, a logic 1 to about two million electrons, a logic 2 to about four million electrons, and a logic 3 to about six million electrons. With presently available lithography, a one-micrometer feature size is possible; CCD processing can yield devices where a logic 1 is about 70,000 electrons.[2]

Because CCD devices consume little power, we can realize complex circuits and increase their density with multiple-valued signals.

The prototype designs achieved at the University of Twente in Holland use more than 70,000 electrons because of the university's lack of suitable lithography equipment. Nonetheless, that value is feasible for a logic 1, because noise in modern CCD circuits does not exceed 2,000 electrons.

CCD has been used extensively in the implementation of analog devices such as delay lines and filters, as well as binary devices such as mass memories. Phillips has produced video memories with 318-kilobit and 835-kilobit capabilities.[2,3] Between the two extremes—analog and binary—comes multiple-valued logic.

We have yet to achieve a natural base, 10, for decimally oriented humans. Presently, 6-logic values are achievable[4]; however, there is much interest in 4-logic levels, since 4 is a power of 2 and thus more compatible with binary circuits. The current limitation on the number of logic levels is not noise, but the chip devices that produce logic values. Deviations in the number of electrons due to variations in the processing of CCD chips exceeds the number due to noise and to charge transfer loss.

The major disadvantage of CCD is speed. CCD uses clock pulses to coordinate the arrival of charge, and the interval between clock pulses must be long enough to complete the slowest operation. Thus, the slowest CCD operation determines the total circuit delay. Binary surface-CCD reportedly has achieved sys-

0018-9162/88/0400-0058$01.00 © 1988 IEEE

tem clock frequencies of up to 40 megahertz.

In spite of the speed disadvantage, CCD is very useful. Compared to all major technologies, CCD has the lowest power/delay dissipation product for a given logic capability. Therefore, even with a larger delay, the power dissipation is so low that the product of the two parameters is smaller than for any other technology. Moreover, we can mitigate the speed disadvantage by using buried-channel CCD, in which charge flows below the surface. Binary buried-channel CCD reportedly has achieved clock frequencies of 180 megahertz.

Current work in CCD has its origins in the 1940s, when K. Schlesinger demonstrated an analog device consisting of a cascade of capacitors along which charge was passed. Two Dutch researchers, F.L.J. Sangster and K. Teer, implemented this *bucket brigade device* as an integrated circuit in 1969. They used MOS technology for the capacitors, with field-effect transistors serving to shepherd the charge through the device.

Most modern CCDs use a charge transfer mechanism proposed by Boyle and Smith[5] in 1970. Called *multiphase clocking*, it operates in a manner analogous to the pouring of water from one cup to another. The efficiency of this mechanism means that little charge is lost in the transfer. Modern CCD circuits, for example, lose only about one in every 10,000 electrons in each charge transfer.[6]

Another improvement in CCD processing followed shortly in 1972. Up to that point, all CCD circuits had the property that the charge resided on the IC surface. It was observed that most of the time spent in the transfer of charge was in the *last* fraction of each charge packet. L.J.M. Esser proposed that this problem be solved by placing the charge well away from the surface. The new devices, called *peristaltic CCD* or *PCCD*, transfer charge at a level below the surface. The result is a higher speed device with an additional benefit: Because charge is away from the surface, electrons in the charge packet are less affected by surface charge recombination and, thus, fewer electrons are lost. (The loss of enough electrons causes a logic level to deteriorate, making it necessary to include logic level regenerators.)

Most electronic devices rely on voltage or current to encode input and output information. Thus, embedded CCD circuits must convert voltage or current to charge and back again. To do this, a capacitor is used. In a capacitor, a linear relationship exists between the charge stored and the voltage across it, given as $q = Cv$, where C is the capacitance. For example, to convert charge to voltage, a charge is placed on a capacitor inducing a voltage that is then amplified and applied to the circuit output.

Input signals can also be entered into a CCD by the absorption of photons. This property was used in 1971 by G.F. Amelio, W.J. Bertram, Jr., and M.F. Tompsett to realize an image reception CCD. The basic principle is used today in commercial cameras and in low-light-level systems, such as telescopic image reception.

The application of CCD to logic circuits was also recognized early. M.F. Tompsett in 1972 showed how to regenerate binary logic levels in memories and proposed the realization of NAND and NOR gates. The first nonbinary CCD circuit was a four-valued memory developed in 1978 by M. Yamada in Japan. The first four-valued CCD implementations of *logic* operations were developed in 1979. In 1980, Kerkhoff[1] constructed arithmetic and higher level four-valued logic operations, as well as two-valued to four-valued and four-valued to two-valued converters.

In the 10 years since the introduction of multiple-valued CCD, there have been advances in three areas—device technology, logic design, and applications. These three areas define the scope of this tutorial. We begin by presenting the principles on which CCD circuits are based. The use of charge to represent logic values determines the way logic is combined to form higher level functions, and we will show its effect on logic design. Then, we will show specific examples of multiple-valued CCD logic design.

Figure 1 shows the hierarchy on which CCD design is based. At the bottom are basic CCD configurations, specifically configurations that are recognized units in the very-large-scale-integration layout and that have a direct effect on movement of charge in the IC. These configurations include storage wells, clocked transfer gates, and barrier gates.

At the next level are logic operations, of which there are four—addition, constant, multithreshold, and inhibit. Each can be described algebraically, and the four together form a *complete* set; that is, a set of operators from which all logic functions can be formed.

The set of *all* logic functions forms the third level of the hierarchy. The focus here is on efficient realizations of given functions. While problems of logic design in binary have been well studied, it is not yet clear how to design multiple-valued circuits. Consequently, this area attracts intense interest. It is interesting to note that the knowledge that the four configurations at the logic operation level are complete comes from the fact that these configurations are sufficient to realize three functions at the combinational logic function level: (1) sum, (2) minimum, and (3) literal, which form a complete set.

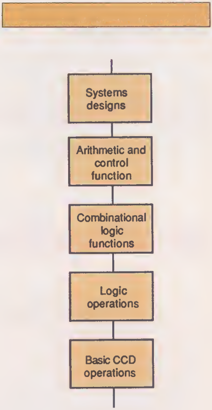

Systems designs
Arithmetic and control function
Combinational logic functions
Logic operations
Basic CCD operations

Figure 1. Hierarchy of CCD systems design.

The next two levels represent large-scale circuits. The fourth level corresponds to arithmetic units, such as adders and multipliers, and control units, such as CPU control units. The highest level corresponds to computer systems and processors. This tutorial focuses on the lowest three levels, these being most affected by IC technology. However, we consider several aspects of the higher two levels.

Fundamentals of CCD logic operations

The basic structure for storing a logic value in conventional logic circuits is a path, which in binary is either open or closed. In CCD, logic values are carried as quantities of charge, and the basic structure for storing a logic value is the *potential well*. The well resembles the capacitor used in nMOS dynamic memories. However, in CCD the capacity of the well is an important parameter. Capacity is measured in logic units, which creates an upper bound on the logic value that can be stored. For example, a well with a capacity of two logic units can hold a logic 0, 1, or 2.

The transfer of charge between wells is accomplished by a *clocked transfer gate*. The time of transfer is determined by clock signals that coordinate the movement of charge around the chip. Figure 2 shows how clocked transfer gates perform addition. Potential wells appear as squares and clocked transfer gates as shaded rectangles separating potential wells. In this case, two clocked transfer gates separate potential well 3 from wells 1 and 2. At the clock pulse, the clocked transfer gates move the two charge units in wells 1 and 2 into well 3, where they are added. The process is analogous to pouring the contents of the input wells into the output well. The transfer is unidirectional, with the direction indicated by context.

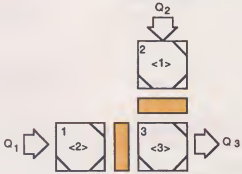


Figure 2. CCD adder.

Basic CCD configurations

Using wells and clocked transfer gates, we can realize higher level functions. Specifically, we discuss four from which it is possible to realize all possible functions. All have been fabricated at the University of Twente, as have all other circuits described in this article.

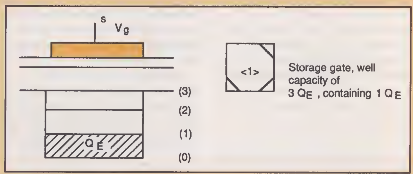
Addition. Because of the fixed size of the well in which the charges are combined, this operation is actually truncated addition. That is, if the sum of the charge put into the sum well exceeds its capacity,

Potential well

The first MVL CCDs were made in a surface CCD process employing double polysilicon interconnections. The IC foundation is a substrate of p-type silicon. Doped silicon-oxide layers are used to establish diffused regions in the substrate and undoped silicon oxide is used for insulation. Two layers of polysilicon form the CCD structures.

The most fundamental CCD structure is the potential well, shown in the accompanying figure. It is a capacitor, one side of which is the substrate and the other an area of polysilicon on top separated by an insulating layer of silicon dioxide.

The well is a region in the substrate where the potential is modified by the voltage applied to the polysilicon. This potential determines the amount of charge the well can contain. The figure shows the side view of a well (left) that can contain a logic 3. A smaller potential would create a well of capacity 2, and a still smaller potential a capacity of 1. The top view (right) shows the potential well as a square. The well's capacity in logic values is indicated by the number of notches in corners. For example, the three notches indicate a capacity of logic 3.


s
V_g
(3)
(2)
(1)
(0)
Q_E
<1>
Storage gate, well capacity of $3\ Q_E$, containing $1\ Q_E$

Potential well.

the excess charge is lost. Thus, the output is the sum of the inputs except where that sum exceeds the sum well capacity *c*, in which case the output is *c*.

Because of its relative simplicity, addition is a widely used operation. It is a key operation in the logic design algorithms discussed later.

Multithreshold. The multithreshold circuit acts like a threshold gate producing an output only when the input exceeds some predetermined value. Figure 3 shows an example of such a circuit. Here there is a single input and three outputs. At the clock pulse, the input charge (to the left of the clocked transfer gate, represented by the vertical shaded bar) flows to wells 1, 3, and 5. Separating these wells are fixed barrier gates represented by nonfilled rectangles.

Like clocked barrier gates, fixed barrier gates facilitate the transfer of charge. Unlike clocked barrier gates, the transfer can occur any time, not just at clock pulses.

The operation of a multithreshold cir-

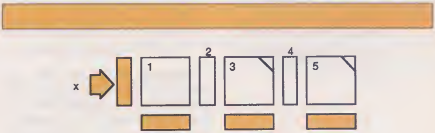


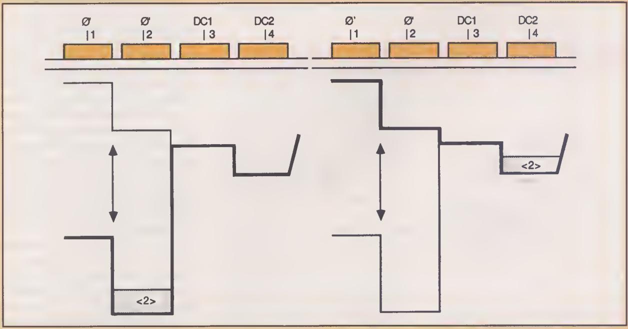
Figure 3. A multithreshold circuit.

Clocked transfer gate

The transfer of charge between two wells is performed in CCD by a gate that drains the first well of its charge, placing it into the second well. A system clock signal activates the transfer.

The top view of a clocked transfer gate is a rectangle, shaded to indicate that it is driven by the system clock. A side view of the gate, as seen in the accompanying figure, shows the voltage values during transfer. The top part of this view shows the two wells and the transfer gate as a set of four capacitors. Two capacitors, 1 and 2, receive a voltage value, ϕ' and ϕ, respectively, that varies between the two values shown below it. The other two capacitors, 3 and 4, receive a constant voltage value.

Prior to the clock pulse (part (a) of the figure), charge resides under capacitor 2, having come in from the left. The clock pulse raises the voltage under capacitors 1 and 2, causing the charge to flow to 4. Part (b) shows the result. The voltage on capacitor 1 is always higher than that of 2, preventing charge from flowing toward the left. For the same reason, the voltage on capacitor 3 is higher than on 4.



Charge transfer in the clocked transfer gate, showing (a) charge before clock pulse, and (b) charge after clock pulse.

cuit resembles the filling of an ice cube tray. That is, charge flows first to well 1. If this is filled to capacity, excess charge flows to well 3. If this fills to capacity, the remaining excess flows to well 5. Thus, wells 1, 3, and 5 produce a logic 1 output if the input logic value is equal to or greater than 1, 2, and 3, respectively. In this way, the multithreshold circuit acts like a threshold element, producing an output depending on whether a well capacity is exceeded or not.

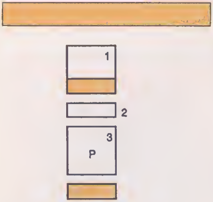


Figure 4. Constant generator.

Constant generator. Figure 4 shows how a constant logic value is produced. Well 1 is a source of charge produced at a clock pulse, as indicated by the shaded bar embedded in well 1. The fixed barrier gate 2, in effect, determines the capacity of well 3, shown here as p. At the clock pulse, well 3 is filled to capacity by the charge source 1. At the next clock pulse, this is transferred out of the constant generator.

Inhibit gate. An important part of the inhibit gate is the floating sense gate. The floating sense gate provides *action-at-a-distance* capability in CCD. It consists of a special potential well called a *sense well* and an attached barrier gate. A wire at the metalization layer connects the sense well and the barrier, so the barrier can be placed away from the sense well.

When the sense well contains a nonzero logic value, the attached barrier gate is high. Conversely, an empty sense well produces a lowered barrier. Figure 5 shows the use of the floating sense gate in an inhibit gate.

The inhibit gate operates as a switch for an input charge. This charge flows out one of two outputs depending on the value of another input, the controlling input. As shown in Figure 5, the main input charge is applied at the top. Barrier gate 3 is high if the controlling input charge in gate 7 is nonzero, so the main input charge in well 2 is inhibited from moving into well 4 (thus the name, inhibit gate); at the clock pulse, it moves right and out of the circuit. On the other hand, if gate 7 has no charge, then barrier gate 3 is down and charge from the top well flows through well 2 across barrier gate 3 and into well 4, where it is clocked out at the lower output.

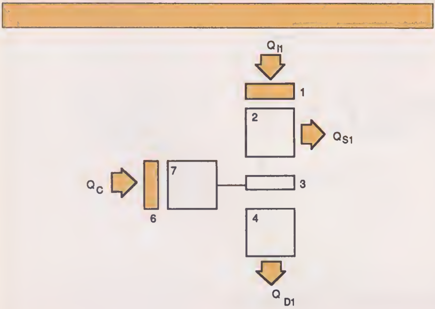


Figure 5. Inhibit gate.

Programmable logic arrays

The capacity of a well is easily changed, since it depends on the applied voltage. As a result, we can "program" the logic function of a CCD circuit by an appropriate choice of voltages. This is the basis of the CCD programmable logic array, or CCD-PLA.[4]

Figure 6 shows the basic CCD-PLA circuit. Here, input x is applied to two multithreshold circuits. In the uppermost multithreshold circuit are two wells, one of capacity a and one of capacity b. At the clock pulse, the charge from x flows right into well a. If $x > a$, then the excess, $x-a$, flows into the adjacent well. At the next clock pulse, $x-a$ in this well flows onto the adder, shown as a line on the right in Figure 6. The diagram to the right of the upper multithreshold circuit shows how the output $x-a$ applied to the adder depends on the input x and a.

The lower multithreshold circuit behaves in the same way except for the complement circuit represented by a triangle. This circuit forms the identical function of the complement circuit shown in Figure 7. However, its implementation in Figure 6 is much simpler. Since x is an external signal, it is encoded as a voltage. It is in the voltage-to-charge conversion that the complement is realized. In fact, the complement is realized with exactly the same circuitry as a noncomplementing voltage-to-charge converter.[1] The diagram to the right of the lower multithreshold circuit shows that its output function is the reverse of that of the upper multithreshold circuit.

In Figure 6, the output is derived from the column that realizes a product term. Figure 7 shows the circuit that the column drives. That is, the column is applied to a sensitive sense amplifier that, in turn, controls the barrier of an inhibit gate consisting of a source well and a well f with programmable capacity f. Thus, when there is charge on the column, the barrier is high, blocking charge from the source

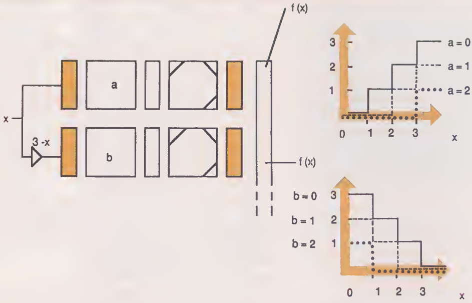


Figure 6. Basic CCD-PLA circuit.

well to well f. When there is no charge, the flow is not blocked and well f fills to its capacity *f*. At the next clock pulse, a quantity of charge, f, is transferred to the output column, where it is summed with the outputs of all other columns.

The output of the PLA column is *f* only if the outputs to all basic CCD-PLA circuits of that column are 0. A basic CCD-PLA produces a 0 when both multithreshold circuits produce a 0. Because of the staircase functions produced by the multithreshold circuits, a 0 is produced only for x in the range between the lowest steps of the two staircase waveforms. For example, if, in Figure 6, $a = 2$ and $b = 2$, then the basic CCD-PLA circuit corresponding to x produces a logic 0 when $1 \leq x \leq 2$ and a logic 1 otherwise.

This logic operation is similar to the following. Let x_i be a four-valued variable. Thus, $0 \leq x_i \leq 3$. The *literal function* is defined as

$$f(x_i) = {}^{a}x_i^{b} = \begin{cases} r-1 \text{ if } a \leq x_i \leq b \\ 0 \text{ otherwise} \end{cases}$$

Since the output values of the literal operation are just 0 and $r-1$, this operation yields binary output values given the full range of multivalued input values. Specifically, the output assumes the higher value only when the input values occur in the range $a \leq x_i \leq b$. It is also called a *decisive* function because it executes a decision operation, placing logic values into two categories, those which produce 0 and those which produce $r-1$.

The *Min operation* is defined as

$$x \cdot y = Min(x, y)$$

The term *min* describes the fact that the output value is the minimum of the two input values. The output of the inhibit controlled by the sum of the basic CCD-PLA circuits will be *f* if there is no charge in the sum and 0 if there is a nonzero charge. Thus, the CCD-PLA column produces a logic function that can be written as

$$f \cdot {}^{a_1}x_1^{b_1} \cdot {}^{a_2}x_2^{b_2} \cdot \;\ldots\; \cdot {}^{a_n}x_n^{b_n}$$

This is called a *product term* because of its resemblance to multiplication. The product term produces logic value *f* if $a_i \leq x_i \leq b_i$ for all i, and produces 0 otherwise.

The output column produces the (truncated) sum of the column outputs, and the resulting sum-of-products is the PLA output. We know that *any* function has at least one sum-of-products expression. Thus, any function can be realized by a CCD-PLA of the appropriate size.

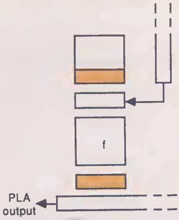


Figure 7. CCD-PLA column output circuit.

Figure 8 shows a photomicrograph of the basic CCD-PLA circuit. The object left-of-center, which is the junction for many leads, is the basic CCD-PLA cell. The two multithreshold circuits occur in

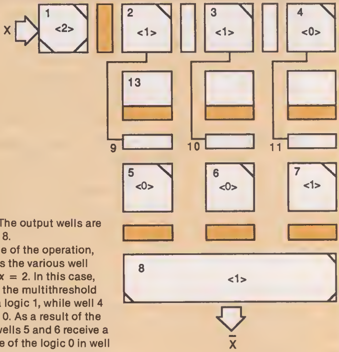
Complement circuit

A binary complement circuit is a single-input, single-output device that realizes the expression $q = (r-1) - x$, where x is the input, q is the output, and r is the radix. When $r = 2$, this corresponds to the binary complement operation $q = \bar{x}$.

The accompanying figure shows a CCD realization of a four-valued complementer ($r = 4$). The circuit consists of two parts, a multithreshold circuit and three inhibit circuits that drive an adder that provides the circuit's output. The multithreshold circuit serves to divide four-valued variable x into three two-valued quantities. Specifically, if x is a logic 3, then a logic 1 exists in each of the wells 2, 3, and 4. If x is a logic 2, then a logic 1 exists in wells 2 and 3 only. If x is a logic 1, then a logic 1 exists only in well 2.

Each of the wells 2, 3, and 4 is a sense well for a barrier gate separating a source of charge from a well of capacity one. Each of these three circuits forms a binary complement, with the barrier passing a logic 1 if the corresponding sense well has a 0 and vice versa. The output wells are summed in well 8.

As an example of the operation, the figure shows the various well contents when $x = 2$. In this case, wells 2 and 3 of the multithreshold circuit receive a logic 1, while well 4 receives a logic 0. As a result of the inhibit action, wells 5 and 6 receive a logic 0. Because of the logic 0 in well 4, the corresponding barrier gate is down and well 7 receives a logic 1. Thus, the output, which is the sum of the wells, is 1.



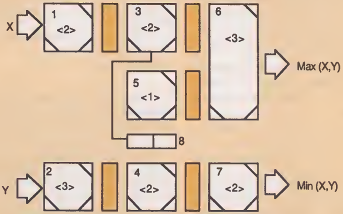
Four-valued complement circuit.

Minimum/maximum circuit

Shown in the accompanying figure is a CCD circuit that realizes the maximum and minimum of two logic values x and y. The circuit, in effect, acts as a switch, placing at *Max*(x, y) the larger of the two values and at *Min*(x, y) the smaller. It employs a unique capability of CCD.

The two inputs x and y appear in wells 1 and 2 initially. 1 and 2 are transferred through the clocked transfer gates shown into wells 3 and 4, respectively. Well 3 is a sense well for a special barrier well 8. That is, the barrier height of 8 is directly proportional to the amount of charge in well 3. Furthermore, this barrier determines the capacity of well 4. Specifically, if well 3 contains logic value x, then the capacity of well 4 is x. Any charge from y in excess of x flows over the barrier into well 5. In this example, $x = 2$ and y = 3. Thus, the capacity of well 4 is two. Since y exceeds this by one, a logic 1 flows over the barrier into well 5. At the clock pulse, the sum of the contents of wells 3 and 5 appears in well 6 while the contents of well 4 are transferred into well 7.

Well 7, *Min*(x, y), contains the minimum of the two values since if $x > y$, the value of y is transferred undiminished into well 7. Conversely, well 6, *Max*(x, y), receives x if $x >$ y and receives $x + y - x =$ y if y $> x$.



The maximum/minimum circuit.

the top and bottom halves of this part of the figure. The large uniform square near the upper right-hand corner of the picture is a capacitor used to measure characteristics of the substrate. To its left and slightly lower is a transistor used to measure threshold voltages of the chip. In the lower right-hand corner is the balanced sense amplifier used in detecting charge on the column output.

It is interesting that the CCD-PLA structure remains unchanged if we choose a larger or smaller radix. The only change needed is in the number of voltages for programming well capacities. In such a circuit, we choose the highest radix needed to provide sufficient separation between logic levels, since this yields the highest logical complexity per unit of chip area. That is, increasing the radix is effectively the same as making the circuit more compact.

Logic design of MVL-CCD

We can combine basic CCD gate configurations to form higher level operations. While basic CCD configurations model closely the physical behavior of the CCD circuit, it is easier to design using the higher level operations. In fact, our knowledge that the four basic CCD operations are sufficient to realize all functions comes from the fact that such operations realize a set of higher level operations that are known to realize all functions. Specifically, the four basic operations can be used to realize the Max, Min, and literal operations, a set we know will realize any logic function. We show two such synthesis techniques for multiple-valued CCD circuits; other techniques are discussed in Kerkhoff.[1] Unlike binary, which has a long history of logic algorithm research, we know considerably less about multiple-valued logic design.

Having defined the four basic CCD configurations and their operation, we are in a position to consider the problem of realizing logic circuits from these configurations. The problem is nontrivial because of the extreme complexity of computing systems. It follows that a design approach must be easily implemented by a computer program and must produce reasonably efficient designs.

One approach introduced by Kerkhoff[1] is called the *cost table* technique because each entry has a cost associated with it. To realize a given function, a selection is made from the table so that the selected functions combine to produce the given function at the lowest possible cost.

The cost table approach combines the advantage of custom design (efficient realizations) with a formal procedure that can be implemented by a computer program. There is also a flexibility associated with the choice of cost. While the presently used costs correlate closely with the IC chip area, other possible costs are (1) delay and (2) yield. For example, in the case of the latter cost, some CCD operations cannot be produced as reliably as others, so a smaller percentage of manufactured circuits will operate within specifications.

Because the realization of a given function can be accomplished by an appropriate choice of functions from the cost table, one synthesis technique is exhaustive enumeration. That is, we simply enumerate all possible ways to realize a given function and then choose the one with the least cost.

While practical for one-input, one-output functions, exhaustive enumeration is prohibitively time-consuming for functions of two or more variables. This observation has inspired research into other approaches, such as directed search, which uses an approximation to the costs used in previous work. The approximation has the advantage that costs are extracted directly from the function, allowing a formal process in which certain realizations of the given function can be quickly eliminated as nonminimum realizations.

The logic design of binary PLAs has received considerable attention. The problem of finding a minimal solution is known to be NP-complete, which has inspired research on heuristic approaches to binary PLA minimization.

A similar emphasis on heuristic approaches to multiple-valued PLAs exists. At present, there are three similar approaches. All are two-step processes. The first step is to select a minterm—an assignment of values to the variables such that the given function is nonzero. For such an assignment, at least one column produces a nonzero value. The second step is to find a product term that covers the selected minterm. This corresponds to choosing the function realized by one column that produces a nonzero value when the selected assignment of input values is applied to the PLA. The product term is subtracted from the function and the process is repeated on the resulting function.

In the first method, used by Pomper and Armstrong,[7] the minterm is selected randomly and the product term is selected as the largest that covers it.

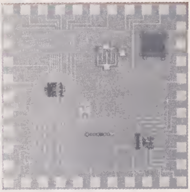


Figure 8. Photomicrograph of the basic CCD-PLA circuit.

In the second method, used by Besslich,[8] the minterm is selected as the most *isolated* minterm—the minterm farthest away from all other minterms. The reduction of the most isolated minterm is an attempt to find an essential product term, a term that appears in all minimal sum-of-product expressions. The product term chosen is one that tends to drive the largest number of minterms to 0 when the selected product term is subtracted.

The third method, used by Dueck and Miller,[9] selects the minterm in a similar way to that of Besslich.[8] However, the product term selected is one that tends to produce the most equal balance among nearby minterms; that is, it tends to choose product terms that when subtracted tend to produce equal logic values in nearby minterms.

All three techniques apply or can be easily modified to apply to the minimization of sum-of-products expressions, where "sum" refers to the truncated sum of the CCD sum operation. Interestingly, the truncated sum operation has been shown to be superior to the Max operation over two classes of multiple-valued logic functions.[10] That is, on the average, fewer product terms are required for the realization of functions when the sum is used compared to when the Max is used. We know also that there are functions for which the Max operation would require significantly fewer products than the sum.

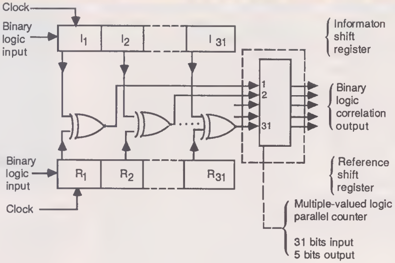


Figure 9. Circuit for a parallel correlator circuit.

MVL CCD-PLA design

In a binary PLA, a given function is realized as the OR of product terms, where a product term is the AND of binary variables or their complement. For example, the two-variable exclusive OR function is realized as the OR of two product terms, $x_1\overline{x}_2 + \overline{x}_1x_2$, where + is OR. In a multiple-valued PLA, the OR function is replaced by the truncated sum (also represented as +), which is normal addition except when the result exceeds $r-1$, the highest logic value, in which case the result is taken as $r-1$.

A product term

$$c \cdot {}^{a_1}x_1{}^{b_1} \cdot {}^{a_2}x_2{}^{b_2} \cdot \ \ldots \ \cdot {}^{a_n}x_n{}^{b_n}$$

in multiple-valued logic is more complicated because of the larger number of logic values. The additional values affect both the product term value and the form of the one-variable functions. The product term value is specified by a constant c, where c is one of the nonzero logic values. The one-variable literal function ${}^{a_i}x_i{}^{b_i}$ provides a discrimination among values of x_i. When x_i is between a_i and b_i, the function is $r-1$, the highest nonzero logic value. Outside this range, the function is 0. When x_i is a binary-valued variable, there are only two nontrivial literal functions, ${}^1x^1 = x$ and ${}^0x^0 = \overline{x}$.

Each column in a CCD-PLA realizes a product term

$$c \cdot {}^{a_1}x_1{}^{b_1} \cdot {}^{a_2}x_2{}^{b_2} \cdot \ \ldots \ \cdot {}^{a_n}x_n{}^{b_n}$$

through a mechanism that depends on the presence or absence of charge deposited on the column. That is, each basic CCD-PLA circuit deposits a nonzero charge on the column if and only if x_i is outside the range a_i through b_i. Thus, a column has no charge if and only if *all* variables are within their specified range.

When no charge is on the column, the inhibit input, which is driven by the column, produces at its output a charge corresponding to a logic c. This is precisely the function

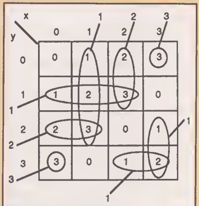
$$c \cdot {}^{a_1}x_1{}^{b_1} \cdot {}^{a_2}x_2{}^{b_2} \cdot \ \ldots \ \cdot {}^{a_n}x_n{}^{b_n}$$

The column output values are all summed to form the PLA output.

The accompanying figure shows how the two-variable function *Modsum*(x, y) is realized as a sum-of-products function of a CCD-PLA. This function is analogous to the exclusive OR function, which is *Modsum*(x, y) in binary logic. The circles in the figure represent product terms whose constant value is shown nearby. For example, the circle surrounding the logic values in the squares $x = 2$ and $0 \leq y \leq 1$ corresponds to the product term $2 \cdot {}^2x^2 \cdot {}^0y^1$. The sum-of-products expression for this realization is

$$\mathit{Modsum}(x, y) = 1 \cdot {}^1x^1 \cdot {}^0y^2 + 1 \cdot {}^0x^2 \cdot {}^1y^1 + 2 \cdot {}^2x^2 \cdot {}^0y^1 + 2 \cdot {}^0x^1 \cdot {}^2y^2 + 3 \cdot {}^0x^0 \cdot {}^3y^3 + 3 \cdot {}^3x^3 \cdot {}^0y^0 + 1 \cdot {}^2x^3 \cdot {}^3y^3 + 1 \cdot {}^3x^{3 \cdot 2}y^3$$

There are six product terms, so a six-column CCD-PLA is needed.



y \ x	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

***Modsum*(x, y) function.**

But then, we know that the converse is also true.

These techniques are a necessary part of CAD tools for the design of multiple-valued PLAs. A CAD tool for the design of CCD-PLAs already exists,[11] the first known tool for multiple-valued logic design.

Applications

As one example of a CCD logic design application, consider a parallel correlator. Such circuits are useful in sonar, radar, and communications systems where an incoming signal is compared with a known signal. If the two are highly correlated, then the incoming signal is judged to be an instance of the known signal. This process allows us to detect weak signals buried in noise.

Figure 9 shows a correlator for a 31-bit binary signal that could, for example, be a discretized version of a radar return signal. The binary circuit input is shifted into the information shift register, while the reference signal is shifted into the reference shift register. The two streams are compared bit-by-bit by 31 exclusive-OR gates. A logic 1 at the output indicates that the two input bits are identical. By counting the number of bits at the output of these gates, a measure of the correlation is obtained. This is done by a parallel counter with 31 inputs and five outputs, where the output is a binary number representing the number of 1's on the inputs. A compact realization of the parallel counter is achieved through the use of four-valued CCD circuits. Figure 10 shows that the counter consists of four 7-bit adders (7BA) and three quaternary full adders (QFA).

Each seven-bit and quaternary full adder produces two outputs, a four-valued output representing the sum and a binary output representing the carry. The input to the quaternary full adder is a pair of four-valued variables and a binary carry input. Each seven-bit binary adder has seven binary inputs and is a quaternary full adder in which the two four-valued inputs are replaced by six binary inputs.

The circuits for the binary and quaternary full adders in CCD have the same complexity, so the circuit shown in Figure 10 uses approximately half the circuitry required by an all-binary design. Also, there is a considerable reduction in interconnections. Since most of the area of typical binary VLSI circuits is devoted to interconnections, this is significant.

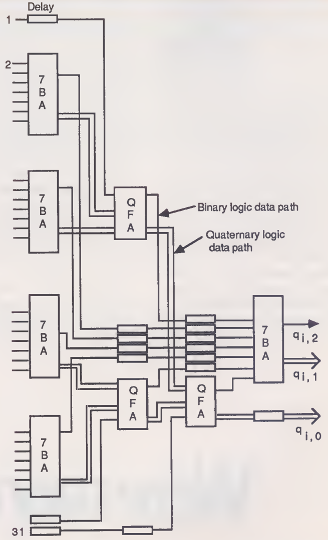


Figure 10. A 31-bit parallel counter using seven-bit adders and quaternary adders.

Another application that CCD suits admirably is in image processing. The use of multiple-valued logic is especially attractive because it eliminates the decoding operation between a gray-level image and the binary representation of that gray level.

Kerkhoff, Zijlstra, and Onneweer[12] showed an architecture for a multivalued image processor using pipelined CMOS and CCD. Unlike conventional designs where image reception and storage are separate, this MVL processor uses the light-reception CCD to store the image. The heart of the processor is a multivalued pixel analyzer that determines subsequent pixel values as a function of pixels in a 3×3 window surrounding the pixel. For example, it is possible to shrink objects to a simple cell, allowing an automated counting technique. Also, alphanumeric characters can be skeletonized to stick-like figures applicable to pattern-recognition programs.

Because little power is consumed in CCD, complex circuits can be realized with very small interdevice distances. The use of multiple-valued signals further improves logic density. While six-valued CCD is possible, four-valued CCD evokes the most interest because of its compatibility with binary circuits.

The realization of four-valued CCD logic circuits is accomplished through four basic CCD configurations. These form a complete set from which any function can be realized. One promising design approach is the cost table technique. Here, a given function is realized by selecting functions from the table and combining them. Another approach is the PLA. This fixed structure CCD circuit is programmed to realize a given function by adjusting well capacities.

The major disadvantages of CCD are speed and the necessity of clock signals. Thus, CCD has application where speed is not as critical as complexity. CCD will continue to be used in image processing, an area where it is the dominant technology. Also, CCD will continue to find application in extremely compact memory circuits. Past successes include 318-kilobit and 856-kilobit video memories by Phillips.[2,3] It is expected that, as logic is merged with memory, there will be increased use of CCD in the implementation of combinational logic.

The cost of CCD processing, which was high 10 years ago compared to the cost of CMOS, is now comparable. There is considerable interest in merging the two technologies, with the prospect of finding a hybrid that takes advantage of the best of both technologies. There is also interest in using CCD in implementing neural networks, where high-speed computation is not so critical as in conventional computing. For example, a serial CCD neural net has been fabricated at the University of Southern California.

The full potential of multiple-valued CCD will require an understanding of the design process. It is not yet clear what basic configurations and what design techniques provide the best trade-off between flexibility and efficiency of realization. The question has been answered for the most part in binary, because of the small number of such operations and because of a long period of research. However, even in three-valued logic there are no definitive answers for the present. □

Acknowledgments

We wish to thank the North Atlantic Treaty Organization, the Dutch Foundation for Fundamental Research on Matter (FOM), and the National Science Foundation for grants that supported research reported in this article.

References

1. H.G. Kerkhoff, *Theory, Design, and Applications of Digital Charge-Coupled Devices*, PhD Thesis, Twente University of Technology, Enschede, The Netherlands, Apr. 1984.
2. M.J.M. Pelgrom et al. "Digital Field Memory for Television Receivers," *IEEE Trans. Consumer Electronics*, Vol. CE-29, Aug.

We're more compatible

1983, pp. 242-250, and personal communications with M.J.M. Pelgrom.

3. H. Veendrick et al., "835Kb CCD Video Serial Memory," *Proc. Int'l Solid State Circuits Conf.*, Feb. 1988.
4. H.G. Kerkhoff and J.T. Butler, "Design of a High-Radix Programmable Logic Array Using Profiled Peristaltic Charge-Coupled Devices," *Proc. 16th Int'l Symp. Multiple-Valued Logic*, May 1986, pp. 128-136.
5. W.S. Boyle and G.E. Smith, "Charge-Coupled Semiconductor Devices," *Bell System Technical J.*, Vol. 49, Apr. 1970, pp. 587-593.
6. H.S. Abdel-Aty Zohdy, S.G. Chamberlain, and L.A.K. Watt, "Limitations of Multilevel Storage in Charge-Coupled Devices," *IEEE Trans. Electronic Devices*, Vol. ED-27, Sept. 1980, pp. 1733-1743.
7. G. Pomper and J.R. Armstrong, "Representation of Multivalued Functions Using the Direct Cover Method," *IEEE Trans. Computers*, Vol. C-30, Sept. 1981, pp. 674-679.
8. P.W. Besslich, "Heuristic Minimization of MVL Functions: A Direct Cover Approach," *IEEE Trans. Computers*, Vol. C-35, Feb. 1986, pp. 134-144.
9. G.W. Dueck and D.M. Miller, "A Direct Cover MVL Minimization Using the Truncated Sum," *Proc. 17th Int'l Symp. Multiple-Valued Logic*, May 1987, pp. 221-227.
10. E.A. Bender, J.T. Butler, and H.G. Kerkhoff, "Comparing the Sum with the Max for use in Four-Valued PLAs," *Proc. 15th Int'l Symp. Multiple-Valued Logic*, May 1985, pp. 30-35.
11. H.G. Kerkhoff and J.T. Butler, "A Module Compiler for High-Radix CCD-PLAs," preprint.
12. H.G. Kerkhoff, P. Zijlstra, and S.P. Onneweer, "A Pipelined MVL Image-Processing System Based on a Combined CCD-CMOS Process," *Proc. 15th Int'l Symp. Multiple-Valued Logic*, May 1985, pp. 361-368.

Jon T. Butler is guest editor of this issue of *Computer*. His biography and photo appear at the end of the Guest Editor's Introduction.

Hans G. Kerkhoff has been a staff member of the IC-Technology and Electronics Group of the University of Twente, The Netherlands, since 1977. He is currently interested in testable computer-aided design of CMOS VLSI dedicated digital signal processors.

Kerkhoff received the MSc degree in electronic engineering from the Technical University of Delft, The Netherlands, in 1977. He obtained the PhD degree in microelectronics design from the University of Twente in 1984.

Readers interested in receiving reprints of papers on multiple-valued CCD should circle Reader Service Number 160 on the reader service card.

Readers may write to Butler (Code 62-Bu) at the Dept. of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA 93943-5100.

with the way you work.

Index Technology is the leader in providing CASE solutions.

Because we work the way you do.

Instead of forcing a new methodology on you, we provide products that can be customized to meet your needs. You can work alone or in a group. And you can choose the style and technique that works best for you.

We don't quit until the job is done, either. Because our Excelerator® family of products supports the entire development process—from strategic planning to analysis and design—even to code generation.

What's more, we're more compatible with the hardware and software that work best for developers of both MIS and real time systems. Excelerator products run on the IBM PC, DEC VAXstation, SUN and Apollo workstations, and link to numerous other software programs.

Call us for a demonstration at 617-494-8244.
Or write us at Index Technology Corporation,
One Main Street, Cambridge, MA 02142.

Index Technology

® Excelerator is a registered trademark of Index Technology Corporation.

Reader Service Number 5

CALL FOR PAPERS

DATA ENGINEERING

Fifth International Conference on Data Engineering

February 7-9, 1989
Los Angeles, California, USA
sponsored by The Computer Society of the IEEE

COMMITTEE

Steering Committee:
C. V Ramamoorthy, University of California, Berkeley
P Bruce Berra, Syracuse University
Gio Wiederhold, Stanford University
Benjamin W Wah, University of Illinois

General Chairperson:
John Carlis, University of Minnesota

Program Chairperson:
Richard L Shuey, Rensselaer Polytechnic Institute

Program Co-Chairpersons:
Mohan Ahuja, Ohio State University (Tutorials)
Ophir Frieder, Bell Communications Research
Sushil Jajodia, National Science Foundation
Iris Kameny, Rand Corporation
Raymond Liuzzi, USAF RADC
Gordon T Ray, NEC America (International Far East)
Gunter Schlageter, University of Hagen (International Europe)
Amit P Sheth, UNISYS
David Spooner, Rensselaer Polytechnic Institute
Marjorie P Templeton, UNISYS
Kyu-Young Whang, IBM Yorktown Res Center

Awards:
David Du, University of Minnesota

Publicity:
James P Held, University of Minnesota

Treasurer:
Douglas K Barry, Control Data Corp

Local Arrangements:
Homideh Afsarmanesh, Calif State Univ D.H.

Industrial Coordinator:
Elmer Baldwin, Oracle Corp

International Coordination:
Tadao Ichikawa, Hiroshima University
Witold Litwin, INRIA

COMMITTEE MEMBERS (TENTATIVE)

Peter Apers
Adarsh K Arora
Eric M Aupperle
J. L Baer
Don Batory
Katharina M Baumgartner
G Belford
Bharat Bhargava
Michael R Blaha
Diana L Bodwin
Richard Braegger
Walter A Burkhard
C. Robert Carlson
Nick Cercone
Peter Chen
Hong-Tai Chou
David Choy
Wesley W Chu
Peter Dadam
George O Detlefsen
Earl F Ecklund, Jr
Ramez ElMasri
David W Embley
Michael Evangelist
Don Fastz
James P Fry
Hector Garcia-Molina
Georges Gardarin
Robert Gerber
Sakti P Ghosh
Mark Gooley
Georg Gottlob
Walter Grattidge
Laura Haas
Martin Hardwick
Lee Hollaar
Yang-Chang Hong
David K Hsiao
Cheng Hsu
Tadao Ichikawa
H. Ishikawa
Hemant K Jain
Jie-Yong Juang
Arthur Keller
Larry Kerschberg
K H. Kim
Dan Kogan
Walter Kohler
Robert R. Korfhage
Tosiyasu L Kunii
Winfried Lamersdorf
James A Larson
Matt LaSaine
Herbert M Lee
Francis Leung
Phillip M. Lewis
Guo-Jie Li
Victor Li
Yao-Nan Lien
Les Lilien
Jane Liu
Michael M. T. Liu
Vincent Lum
Stephen F Lundstrom
Yuen-Wah Ma
Mamoru Maekawa
Fred Maryanski
Gordon McCalla
John McCarthy
Toshimi Minoura
J Eliot Moss
Tadao Murata
Sham Navathe
Philip M. Neches
Richard E. Newman-Wolfe
Ole Oren
Z. Meral Ozsoyoglu
Christine Parent
J. F. Paris
Krithi Ramamritham
K. V S. Ramarao
Peter Rathmann
David Reiner
Alain F Robert
Gruia-Catalin Roman
Domenico Sacca
Giovanni Maria Sacco
Sharon Salveter
Philip Sheu
Edgar Sibley
David Stemple
Peter M Stocker
Michael Stonebraker
Stanley Y. W Su
Denji Tajima
T. C. Ting
A. M. Tjoa
Mas Tsuchiya
Yosihisa Udagawa
Joseph Urban
Susan D. Urban
Patrick Valduriez
Ralph F. Wachter
Jack H Westbrook
Chao-Chin Yang
S Bing Yao
Clement Yu

SCOPE

Data Engineering is concerned with the semantics and structuring of data in information system design, development, management, and use; and with computer system and architectural considerations that are relevant to that concern. It encompasses both traditional and emerging issues and applications. The purpose of this conference is to provide a forum for the sharing of practical experiences and research advances from an engineering point of view among those interested in automated data and knowledge management. Our expectation is that this sharing will enable future information systems to be more efficient and effective, and future research to be more relevant and timely.

We are particularly soliciting industrial, business, and government participation. We know it is vital that there be a dialogue between practitioners and researchers. We look forward to reports of information systems experience detailing experiments, evaluation, problems, and opportunities associated with design, implementation, and operation. Such reports will be given special consideration.

TOPICS OF INTEREST INCLUDE

AI and Knowledge Based Systems
Applications and Application Systems
Autonomous Distributed Systems
Communication Systems
Concurrency Control and Data Integrity
Data Access Control and Security
Database Management and Structures
Data Engineering Techniques and Tools
Data Services and Servers
Information System Architecture
Performance Evaluation
User Interfaces

PAPER SUBMISSIONS

Each paper's length should be limited to 8 proceedings pages, which is about 5000 words, or 25 double spaced typed pages. Five copies of completed papers should be mailed before June 15, 1988 to:

Richard L. Shuey, Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, 12189-3590; (518) 276-8376 or (518) 374-5684; Shuey%MTS@ITSGW.RPI.EDU or shuey@ge-crd.arpa

TUTORIALS

The day preceding the conference will be devoted to introductory tutorials which may provide background for the conference proper. The day following the conference will be devoted to advanced tutorials. Proposals for tutorials on Data Engineering topics are welcome. Send proposals by June 15, 1988 to:

Mohan Ahuja, Department of Computer Science, The Ohio State University, 2036 Neil Avenue, Columbus, OH 43210-1277; (614) 292-6377; ahuja@ohio-state.arpa

CONFERENCE TIMETABLE AND INFORMATION

Papers due:	June 15, 1988
Tutorial Proposals Due:	June 15, 1988
Acceptance Letter Sent:	September 15, 1988
Camera Ready Copy Due:	November 1, 1988
Tutorials:	February 6 and 10, 1989
Conference:	February 7-9, 1989

For further information contact the General Chairperson, John Carlis, Computer Science Department, University of Minnesota, 207 Church Street SE, Minneapolis, MN 55455 (612) 625-6092; carlis%umn-cs.arpa@relay.cs.net

AWARDS, STUDENT PAPERS AND SUBSEQUENT PUBLICATION:

Awards will be given to the best paper and to the best student paper (denoted as such when submitted solely by students). The latter will receive the K. S. Fu award honoring one of the early supporters of the conference. Up to three grants of $500.00 each will be available to help defray travel costs of student authors. Outstanding papers will be considered for publication in the IEEE Computer Society publications: Computer, Expert, Software, and Transactions on Software Engineering, etc. For more information, contact the general chairman.

THE COMPUTER SOCIETY OF THE IEEE

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.
IEEE

Multiple-Valued Logic and Optimization of Programmable Logic Arrays

Tsutomu Sasao
Kyushu Institute of Technology

Multiple-valued input, two-valued output functions simplify the task of optimizing programmable logic arrays with input decoders.

Since their introduction in 1970, programmable logic arrays (PLAs) have been used in large-scale-integration/very-large-scale-integration (LSI/VLSI) chips for pocket calculators, watches, and toys. Because PLAs have a regular structure, their design, test, and modification are quite easy. Modern 32-bit microprocessors such as the Motorola MC68020, Intel 80386, AT&T WE32100, and IBM Micro/370 use PLAs extensively in their control sections. Minimizing the number of columns in a PLA is important because it reduces not only the chip area but also the propagation delay and power dissipation.

Multiple-valued logic is used to design two-valued PLAs. A *standard PLA* (or a two-level PLA) has a memory-like structure, as shown in Figure 1. It realizes switching functions in sum-of-products form. Figure 2 shows an example of a standard PLA. The vertical lines realize the products (ANDs) of the input variables or their complements, and the horizontal line realizes the sum (OR) of the products. We can reduce the size of the PLAs by adding decoders to their input. A PLA with two-bit input decoders (see Figure 3a) is an extension of the standard PLA.[1] Figure 3b shows a schematic diagram of a two-bit input decoder. The PLA with two-bit decoders in Figure 4 realizes the same function as the standard PLA in Figure 2, but the PLA with two-bit decoders requires only three columns while the standard PLA requires seven columns. In general, a PLA with two-bit decoders requires 10 to 20 percent fewer columns than a standard PLA.

This article will show a method of designing PLAs using *multiple-valued input, two-valued output functions* (MVITVOFs). A MVITVOF is an extension of the two-valued logic function. An expression for a MVITVOF directly represents a multiple-output PLA with decoders. Each product of the expression corresponds to each column of the PLA, so the number of products in the expression equals the number of columns of the PLA. The array size of the PLA is proportional to the number of products. Thus, we can minimize the PLA by minimizing the expression. Without a MVITVOF, minimizing multiple-output PLAs with decoders is difficult.

New optimization techniques for PLAs

Two new optimization techniques are the (1) assignment of the input variables to the decoders, and (2) output phase optimization. As an example of the first technique, consider the PLA in Figure 4. The variables x_1 and x_2 are connected to the first decoder, and x_3 and x_4 are connected to the second decoder. However, as shown in Figure 5, if x_1 and x_3 are connected to the first decoder, and x_2 and x_4 are connected to the second decoder, then the PLA requires only two columns to realize the same function. The specific assignment of the input variables to the decoders influences the size of the PLA. As an example of output phase optimization, consider the PLA in Figure 6. We can reduce this PLA by realizing the complement of f_0 and adding an inverter to the output f_0, as shown in Figure 7.

0018-9162/88/0400-0071$01.00 © 1988 IEEE

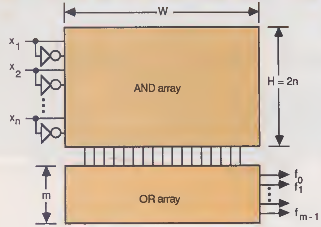


Figure 1. Standard PLA.

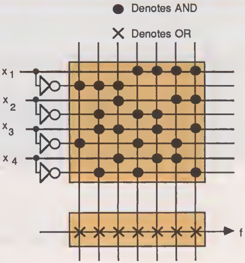


Figure 2. Standard PLA for table "Switching function" in sidebar "Example of PLAs with two-bit decoders."

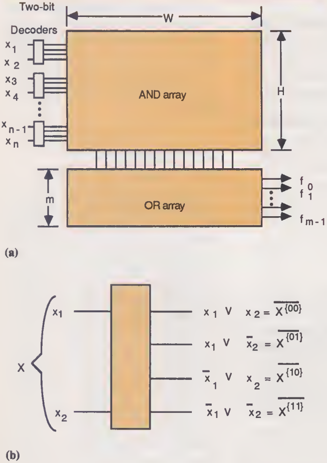

Two-bit
Decoders
W
x_1
x_2
x_3
x_4
x_{n-1}
x_n
AND array
H
m
OR array
f_0
f_1
f_{m-1}
(a)
X
x_1
x_2
$x_1 \vee x_2 = \overline{X^{\{00\}}}$
$x_1 \vee \bar{x}_2 = \overline{X^{\{01\}}}$
$\bar{x}_1 \vee x_2 = \overline{X^{\{10\}}}$
$\bar{x}_1 \vee \bar{x}_2 = \overline{X^{\{11\}}}$
(b)

Figure 3. (a) PLA with two-bit decoders.
(b) Two-bit decoder.

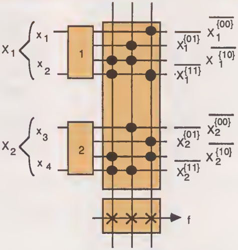

X_1
x_1
x_2
1
$\overline{X_1^{\{00\}}}$
$\overline{X_1^{\{01\}}}$
$\overline{X_1^{\{10\}}}$
$\overline{X_1^{\{11\}}}$
X_2
x_3
x_4
2
$\overline{X_2^{\{00\}}}$
$\overline{X_2^{\{01\}}}$
$\overline{X_2^{\{10\}}}$
$\overline{X_2^{\{11\}}}$
f

Figure 4. PLA with two-bit decoders for table "Switching function" in sidebar "Example of PLAs with two-bit decoders."

A standard PLA consists of the input inverters, the AND array, and the OR array, as shown in Figure 1. The *array size* of the PLA is defined as $W \cdot (H+m)$, where W is the number of columns of the PLA, H is the number of literal lines, and m is the number of outputs.

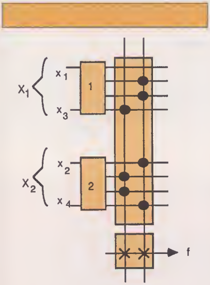


Figure 5. PLA with two-bit decoders for table "Switching function" in sidebar "Example of PLAs with two-bit decoders."

PLAs with decoders

When we replace the inverters of a standard PLA with decoders, we have a *PLA with decoders* (see Figures 3a and 3b).

In general, a t-bit decoder has 2^t outputs and produces all the maxterms of the input variables. The standard PLA is a special case of a PLA with decoders, being a PLA with one-bit decoders.

A PLA with t-bit decoders is represented by an expression of a MVITVOF, $F: P_1 \times P_2 \times \ldots \times P_n \rightarrow \{0, 1\}$, where $P_i = \{0, 1, \ldots, p-1\}$, and $p = 2^t$. When $t = 1$, F is an ordinary two-valued logic function. A minimum sum-of-products expression corresponds to a minimum two-level AND-OR network, or a minimum standard PLA. When $p = 2^t$, where $t \geq 2$, F represents a PLA with t-bit decoders, and a minimum sum-of-products expression for F corresponds to a minimum PLA with t-bit decoders.[2]

PLAs with decoders never require more columns than standard PLAs. In the case of a PLA with two-bit decoders, the number of literal lines equals that of a standard PLA. So, the array size of a PLA with two-bit decoders never exceeds that of a stan-

Table 1. Number of columns to realize n-variable functions by PLAs (n is even).

	Standard PLA	PLA with Two-Bit Decoders
Arbitrary Function (worst case)	2^{n-1}	2^{n-2}
Symmetric Function (worst case)	2^{n-1}	$3^{(n-2)/2}$
Parity Function (worst case)	2^{n-1}	$2^{(n-2)/2}$
Random Function of 10-Variable (average)	163	120

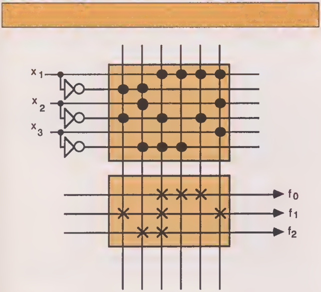


Figure 6. Standard PLA for table "Three-input three-output function" in sidebar "MVITVOF and its expressions."

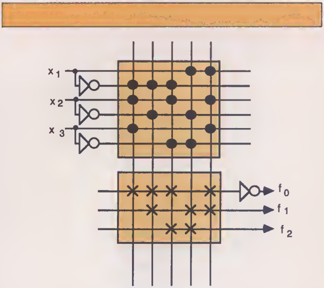


Figure 7. Output phase optimized PLA for table "Three-input three-output function" in sidebar "MVITVOF and its expressions."

dard PLA. Table 1 shows the number of columns needed to realize various classes of functions. When the function has a symmetric property, PLAs with two-bit decoders require much smaller arrays than standard PLAs. However, in the case of a PLA with t-bit decoders, where $t \geq 3$, the number of literal lines is greater than that of a standard PLA. So, a PLA with t-bit decoders ($t \geq 3$) can be larger than a standard PLA even if the number of columns is smaller.

The cost of the decoders is excluded from the PLA's array size; it depends on the technology of the circuit realizing the decoder as well as the size of the PLA. In a dynamic complementary metal oxide semiconductor (CMOS) PLA with 200 columns, the area for one-bit decoders is equivalent to 14 columns while the area for two-bit decoders is equivalent to 29 columns. The cost of a three-bit decoder is at least three times greater than that of a two-bit decoder for a CMOS realization. Thus, we have to add the area for the decoders to compare the actual chip size.

Assignment of input variables to decoders

The specific assignment of the input variables to the decoders often influences the size of the PLA with decoders. An optimum assignment minimizes the size. We can find the optimum assignment by considering all possible assignments. For a PLA with two-bit decoders, the number of combinations is $(n!)/(2^k \cdot k!)$, where $n = 2k$ is the number of input variables. When $n = 8$, the number of combinations is 105, and it is possible to obtain the optimum solution. I minimized many randomly generated functions of eight variables and obtained statistical data.[2] When the assignment of the input variables is nonoptimized, the array size of PLAs with two-bit decoders is 24 percent smaller than that of standard PLAs. When the assignment is optimum, the array size is 32 percent smaller.

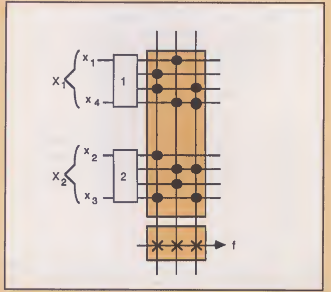
The exhaustive method takes too much computation time for functions with many input variables, so I have developed a heuristic algorithm that quickly obtains good solutions.[3] Table 2 shows the number of

(Continued on p. 78)

Example of PLAs with two-bit decoders

First, we will represent the switching function shown in the table below by using the conventional two-valued logic. Let $X_1 = (x_1)$, $X_2 = (x_2)$, $X_3 = (x_3)$, and $X_4 = (x_4)$ be variables of the MVITVOF. The switching function can be represented as follows:

$$\begin{aligned} f(X_1, X_2, X_3, X_4) = {} & X_1^{\{0\}}X_2^{\{0\}}X_3^{\{0\}}X_4^{\{0\}} \vee X_1^{\{0\}}X_2^{\{0\}}X_3^{\{0\}}X_4^{\{1\}} \vee \\ & X_1^{\{0\}}X_2^{\{0\}}X_3^{\{1\}}X_4^{\{0\}} \vee X_1^{\{0\}}X_2^{\{1\}}X_3^{\{0\}}X_4^{\{0\}} \vee \\ & X_1^{\{0\}}X_2^{\{1\}}X_3^{\{0\}}X_4^{\{1\}} \vee X_1^{\{0\}}X_2^{\{1\}}X_3^{\{1\}}X_4^{\{1\}} \vee \\ & X_1^{\{1\}}X_2^{\{0\}}X_3^{\{0\}}X_4^{\{0\}} \vee X_1^{\{1\}}X_2^{\{0\}}X_3^{\{1\}}X_4^{\{1\}} \vee \\ & X_1^{\{1\}}X_2^{\{1\}}X_3^{\{0\}}X_4^{\{1\}} \vee X_1^{\{1\}}X_2^{\{1\}}X_3^{\{1\}}X_4^{\{0\}} \end{aligned} \tag{1}$$



PLA with two-bit decoders for table "Switching function."

Note that in the conventional two-valued notation, $X^{\{0\}}_i$ is written more concisely as $\overline{X}_i$, and $X^{\{1\}}_i$ as X_i.

Equation 1 shows that f is equal to one if $\{X_1 = 0, X_2 = 0, X_3 = 0$, and $X_4 = 0\}$ or $\{X_1 = 0, X_2 = 0, X_3 = 0$, and $X_4 = 1\}$ or . . . or $\{X_1 = 1, X_2 = 1, X_3 = 1$, and $X_4 = 0\}$. Equation 1 can be simplified as follows:

$$\begin{aligned} f(X_1, X_2, X_3, X_4) = {} & X_1^{\{0\}}X_2^{\{0,1\}}X_3^{\{0\}}X_4^{\{0,1\}} \vee X_1^{\{0\}}X_2^{\{0\}}X_3^{\{1\}}X_4^{\{0\}} \vee \\ & X_1^{\{0\}}X_2^{\{1\}}X_3^{\{1\}}X_4^{\{1\}} \vee X_1^{\{1\}}X_2^{\{0\}}X_3^{\{0\}}X_4^{\{0\}} \vee \\ & X_1^{\{1\}}X_2^{\{0\}}X_3^{\{1\}}X_4^{\{1\}} \vee X_1^{\{1\}}X_2^{\{1\}}X_3^{\{0\}}X_4^{\{1\}} \vee \\ & X_1^{\{1\}}X_2^{\{1\}}X_3^{\{1\}}X_4^{\{0\}} \end{aligned} \tag{2}$$

Equation 2 shows that f becomes one if $\{X_1 = 0$ and $(X_2 = 0$ or $X_2 = 1)$ and $X_3 = 0$ and $(X_4 = 0$ or $X_4 = 1)\}$ or . . . or $\{X_1 = 1, X_2 = 1, X_3 = 1$, and $X_4 = 0\}$. Equation 2 is essentially the same as for an ordinary switching function. Figure 2 (main text) is a realization of the standard PLA for the table at left. In this PLA, each column of the AND array corresponds to each product of Equation 2.

Next, we will consider the four-valued input, two-valued output function. Let $X_1 = (x_1, x_2)$ and $X_2 = (x_3, x_4)$ be the variables of the function. Note that X_1 and X_2 are super-variables and take 00, 01, 10, or 11. $X_1 = 00$ means $x_1 = 0$ and $x_2 = 0$ at the decoder with X_1, and so on.

The expression for the function in the table is

$$\begin{aligned} f(X_1,X_2) = {} & X_1^{\{00\}}X_2^{\{00\}} \vee X_1^{\{00\}}X_2^{\{01\}} \vee X_1^{\{00\}}X_2^{\{10\}} \vee X_1^{\{01\}}X_2^{\{00\}} \vee \\ & X_1^{\{01\}}X_2^{\{01\}} \vee X_1^{\{01\}}X_2^{\{11\}} \vee X_1^{\{10\}}X_2^{\{00\}} \vee X_1^{\{10\}}X_2^{\{11\}} \vee \\ & X_1^{\{11\}}X_2^{\{01\}} \vee X_1^{\{11\}}X_2^{\{10\}} \end{aligned} \tag{3}$$

Equation 3 shows that f becomes one if $\{X_1 = 00$ and $X_2 = 00\}$ or $\{X_1 = 00$ and $X_2 = 01\}$ or . . . or $\{X_1 = 11$ and $X_2 = 10\}$. Note that

$$\begin{aligned} X_1^{\{00\}}X_2^{\{00\}} \vee X_1^{\{00\}}X_2^{\{01\}} &= X_1^{\{00\}} \cdot (X_2^{\{00\}} \vee X_2^{\{01\}}) \\ &= X_1^{\{00\}}X_2^{\{00,01\}} \text{ etc.} \end{aligned} \tag{4}$$

Table 2. Number of columns for arithmetic PLAs.

	Input Data			Standard PLA		PLA with Decoders Worst Input Assignment	PLA with Decoders Optimum Input Assignment	
Function	# In n	# Out m	# Cube	Output Phase Originally	Output Phase Optimized	Output Phase Originally	Output Phase Originally	Output Phase Optmized
$X+Y$	8	5	255	75	61	63	17	14
Log (X)	8	8	255	123	111	113	93	89
$X*Y$	8	8	225	121	108	107	86	73
SQRT (X^2+Y^2)	8	5	255	120	101	101	70	64
$5X+1$(mod 256)	8	8	255	76	76	59	47	47
SQRT (X)	8	5	255	57	48	52	37	32
X^2	8	16	255	180	165	159	142	134
ΣXi	8	4	255	255	186	54	54	38

Equation 3 is minimized to

$$f(X_1,X_2) = X_1^{\{00,01\}}X_2^{\{00,01\}} \vee X_1^{\{00,11\}}X_2^{\{01,10\}} \vee X_1^{\{01,10\}}X_2^{\{00,11\}} \qquad (5)$$

Equation 5 shows that f becomes one if $\{(X_1 = 00$ or $01)$ and $(X_2 = 00$ or $11)\}$, or $\{(X_1 = 00$ or $11)$ and $(X_2 = 01$ or $10)\}$, or $\{(X_1 = 01$ or $10)$ and $(X_2 = 00$ or $11)\}$. Figure 4 (main text) shows the PLA with two-bit decoders, where the AND array corresponds to Equation 5. In fact, the first column realizes the first product because

$$\overline{X_1^{\{10\}}}\cdot\overline{X_1^{\{11\}}}\cdot\overline{X_2^{\{10\}}}\cdot\overline{X_2^{\{11\}}} = X_1^{\{00,01\}}\cdot X_2^{\{00,01\}} \qquad (6)$$

In a similar way, the second column realizes the second product, and so on. Note that the PLA with two-bit decoders uses only three columns, while the standard PLA uses 7 columns.

Assignment of the input variables

Consider a PLA with two-bit decoders realizing the switching function in the table. There are three ways to assign four input variables to two two-bit decoders.

(1) When the input variables are assigned as $X_1 = (x_1, x_2)$ and $X_2 = (x_3, x_4)$ (see Figure 4), the minimum sum-of-products expression is

$$f(X_1, X_2) = X_1^{\{00,01\}}X_2^{\{00,01\}} \vee X_1^{\{00,11\}}X_2^{\{01,10\}} \vee X_1^{\{01,10\}}X_2^{\{00,11\}} \qquad (7)$$

Three columns are necessary in this assignment.

(2) When the input variables are assigned as $X_1 = (x_1, x_3)$ and $X_2 = (x_2, x_4)$ (see Figure 5 in main text), the minimum sum-of-products expression is

$$f(X_1, X_2) = X_1^{\{00,01,10\}}X_2^{\{00,11\}} \vee X_1^{\{00,11\}}X_2^{\{01,10\}} \qquad \cdot(8)$$

Two columns are necessary in this assignment.

(3) When the input variables are assigned as $X_1 = (x_1, x_4)$ and $X_2 = (x_2, x_3)$ (see the figure at left), the minimum sum-of-products expression is

$$f(X_1, X_2) = X_1^{\{00,11\}}X_2^{\{01,10\}} \vee X_1^{\{01,10\}}X_2^{\{00,11\}} \vee X_1^{\{00,01\}}X_2^{\{00,10\}} \qquad (9)$$

Three columns are necessary in this assignment.

Therefore, when we assign the input variables as shown in Figure 5, the array is minimized.

Switching function.

x_1	x_2	x_3	x_4	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

MVITVOF and its expressions

The table below shows an example of a MVITVOF, where X_1 takes two-values, X_2 takes four-values, and X_3 takes three-values. Like ordinary two-valued logic functions, MVITVOFs can be represented by expressions. The function in the truth table is represented as a sum-of-products expression consisting of minterms:

$$F(X_1, X_2, X_3) = X_1^{\{0\}}X_2^{\{0\}}X_3^{\{1\}} \vee X_1^{\{0\}}X_2^{\{1\}}X_3^{\{1\}} \vee X_1^{\{0\}}X_2^{\{2\}}X_3^{\{2\}} \vee X_1^{\{1\}}X_2^{\{0\}}X_3^{\{0\}} \vee X_1^{\{1\}}X_2^{\{0\}}X_3^{\{1\}} \vee X_1^{\{1\}}X_2^{\{0\}}X_3^{\{2\}} \vee X_1^{\{1\}}X_2^{\{1\}}X_3^{\{0\}} \vee X_1^{\{1\}}X_2^{\{2\}}X_3^{\{0\}} \vee X_1^{\{1\}}X_2^{\{3\}}X_3^{\{1\}} \quad (10)$$

Equation 10 shows that F becomes one when $\{X_1 = 0, X_2 = 0,$ and $X_3 = 1\}$ or $\{X_1 = 0, X_2 = 1,$ and $X_3 = 1\}$ or . . . or $\{X_1 = 1, X_2 = 3,$ and $X_3 = 1\}$. Equation 10 is minimized by using the map for the truth table of MVITVOF (see figure at lower left):

$$F(X_1, X_2, X_3) = X_1^{\{0\}}X_2^{\{0,1\}}X_3^{\{1\}} \vee X_1^{\{0\}}X_2^{\{2\}}X_3^{\{2\}} \vee X_1^{\{1\}}X_2^{\{0\}}X_3^{\{0,1,2\}} \vee X_1^{\{1\}}X_2^{\{0,1,2\}}X_3^{\{0\}} \vee X_1^{\{1\}}X_2^{\{3\}}X_3^{\{1\}} \quad (11)$$

Equation 11 shows that F becomes one when $\{X_1 = 0$ and $(X_2 = 0$ or $X_2 = 1)$ and $X_3 = 1\}$ or $\{X_1 = 0, X_2 = 2,$ and $X_3 = 2\}$ or . . . or $\{X_1 = 1, X_2 = 3,$ and $X_3 = 1\}$.

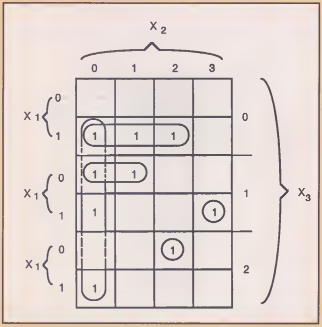
Contact networks with multipositional switches

Expressions for two-valued input logic functions represent contact networks with two-positional switches. In a similar way, expressions for MVITVOFs represent contact networks with multipositional switches.[6] For example, the contact network with multipositional switches shown in the figure below corresponds to Equation 11. In the top row of the network, two-positional switches correspond to X_1; in the second row, four-positional switches correspond to X_2; and in the bottom row, three-positional switches correspond to X_3. Note that the first column of the network realizes the first product of Equation 11. There is a path in the first column only if $\{(X_1 = 0)$ and $(X_2 = 0$ or $X_2 = 1)$ and $(X_3 = 1)\}$. Similarly, the second column realizes the second product, and so on. There is a path between terminals A and B when the value of the expression is one.

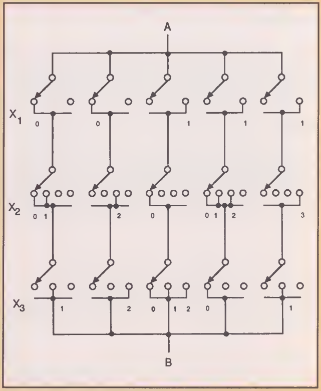
The number of products in the sum-of-products expression is equal to the number of columns of the contact network. Furthermore, a minimum sum-of-products expression corresponds to a contact network with a minimum number of columns.

Truth table of MVITVOF.

X_1	X_2	X_3	F
0	0	0	0
0	0	1	1
0	0	2	0
0	1	0	0
0	1	1	1
0	1	2	0
0	2	0	0
0	2	1	0
0	2	2	1
0	3	0	0
0	3	1	0
0	3	2	0
1	0	0	1
1	0	1	1
1	0	2	1
1	1	0	1
1	1	1	0
1	1	2	0
1	2	0	1
1	2	1	0
1	2	2	0
1	3	0	0
1	3	1	1
1	3	2	0



Map for table "Truth table of MVITVOF."



Contact network with multipositional switches.

An example of multiple-output PLA

Consider the three-input, three-output function shown in the table at upper right, and the characteristic function of the multiple-output function shown in the table below it. Note that the variable *X* representing the outputs takes three values. The minimized expression for the table "Characteristic function" is

$$\begin{aligned} f(X_1, X_2, X_3, X_4) = {} & X_1^{\{0\}}X_2^{\{0\}}X_3^{\{0,1\}}X_4^{\{1\}} \vee X_1^{\{0\}}X_2^{\{1\}}X_3^{\{0\}}X_4^{\{2\}} \vee \\ & X_1^{\{1\}}X_2^{\{0\}}X_3^{\{0\}}X_4^{\{0,1,2\}} \vee X_1^{\{1\}}X_2^{\{0,1\}}X_3^{\{0\}}X_4^{\{0\}} \vee \\ & X_1^{\{1\}}X_2^{\{0\}}X_3^{\{0,1\}}X_4^{\{0\}} \vee X_1^{\{1\}}X_2^{\{1\}}X_3^{\{1\}}X_4^{\{1\}} \end{aligned} \tag{12}$$

Equation 12 shows the same permitted combinations as the characteristic function. Figure 6 (main text) shows a multiple output PLA realizing this expression. Each column including the OR part of the PLA corresponds to a product of the expression, so the number of columns of a PLA is equal to the number of the products in a sum-of-products expression for the characteristic function.

Multiple-output PLA with decoders

An expression of a MVITVOF represents a multiple-output PLA with decoders. For example, consider the realization of the three-input, three-output function (see table at upper right) by using a PLA with a one-bit decoder and a two-bit decoder. Let $X_1 = (x_2)$ and $X_2 = (x_2, x_3)$ denote the input variables, and X_3 denote the outputs. Note that X_1, X_2, and X_3 take two, four, and three values, respectively. The truth table of the characteristic function is the same as the truth table of MVITVOF, except that in X_2, 0 is replaced by 00, 1 by 01, 2 by 10, and 3 by 11. Therefore, the minimized expression obtained from Equation 11 is

$$\begin{aligned} F(X_1, X_2, X_3) = {} & X_1^{\{0\}}X_2^{\{00,01\}}X_3^{\{1\}} \vee X_1^{\{0\}}X_2^{\{10\}}X_3^{\{2\}} \vee \\ & X_1^{\{1\}}X_2^{\{00\}}X_3^{\{0,1,2\}} \vee X_1^{\{1\}}X_2^{\{00,01,10\}}X_3^{\{0\}} \vee \\ & X_1^{\{1\}}X_2^{\{11\}}X_3^{\{1\}} \end{aligned} \tag{13}$$

The figure at right shows the PLA with a one-bit and a two-bit decoder.

Minimization of expressions for MVITVOF

Minimization algorithms for expressions of MVITVOFs[2,4,6,7] are similar to the Quine-McCluskey method for two-valued input logic functions. The major difference is that the input variables may take on more than two values. MVITVOFs with "don't cares" can be treated similarly to two-valued cases. For multiple-valued functions, the number of prime implicants is much larger than that of the two-valued input functions. Therefore, minimization of expressions for MVITVOFs is more time-consuming than for two-valued cases.

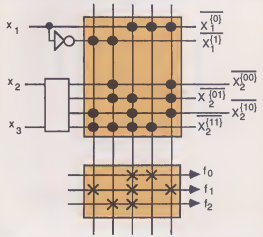
It is impractical to find absolute minimum solutions for large problems because computation time and storage requirements for the minimization programs increase exponentially with the number of prime implicants. MINI,[8] MINI-II,[3] and ESPRESSO-MV[4] are heuristic programs that obtain near-minimum sum-of-products expressions for MVITVOFs. In these programs, expand, reduce, and other operations iteratively simplify the expressions. Because the expand is the most time-consuming operation in the heuristic programs, a complement of a MVITVOF is used to perform the expand operation efficiently.[9]

Three-input, three-output function.

x_1	x_2	x_3	f_0	f_1	f_2
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	1	0

Characteristic function.

X_1	X_2	X_3	X_4	F
0	0	0	0	0
0	0	0	1	1
0	0	0	2	0
0	0	1	0	0
0	0	1	1	1
0	0	1	2	0
0	1	0	0	0
0	1	0	1	0
0	1	0	2	1
0	1	1	0	0
0	1	1	1	0
0	1	1	2	0
1	0	0	0	1
1	0	0	1	1
1	0	0	2	1
1	0	1	0	1
1	0	1	1	0
1	0	1	2	0
1	1	0	0	1
1	1	0	1	0
1	1	0	2	0
1	1	1	0	0
1	1	1	1	1
1	1	1	2	0


x_1
x_2
x_3
$\overline{X_1^{\{0\}}}$
$\overline{X_1^{\{1\}}}$
$\overline{X_2^{\{00\}}}$
$\overline{X_2^{\{01\}}}$
$\overline{X_2^{\{10\}}}$
$\overline{X_2^{\{11\}}}$
f_0
f_1
f_2

PLA with decoders for table "Three-input, three-output function."

(Continued from p. 74)

columns for various arithmetic PLAs. I have optimized these PLAs by exhaustive and heuristic methods. I also designed 16 control circuits for microprocessors. When we optimize the assignment of the input variables, the array size of PLAs with two-bit decoders is, on the average, 20 percent smaller than that of standard PLAs.[3]

We can also consider a PLA with multibit decoders, where each decoder may

Multiple-valued PLAs and encoding problems

A four-valued PLA adder[11] (see figure below) consists of literal generators, the AND array, the OR array, and programmable output encoders. In this PLA, external input and output lines take four different physical states, say 0, 1, 2, and 3 volts. However, the body of the PLA is two-valued and takes only 0 and 3 volts.

The literal generator (see table) converts a four-valued signal into two-valued signals. The CMOS realization of the literal generator (see facing figure) uses inverters having three different thresholds (see table).

The programmable output encoders convert two-valued signals into four-valued ones, as shown in the tables "Output encoding for Sum" and "Output encoding for Carry." Note that the encodings for Sum and Carry are different. In a CMOS realization of a programmable output encoder (see the facing figure) four different voltage sources (0, 1, 2, and 3 volts) are connected to C_0, C_1, C_2, and C_3. We can permute the voltage sources to have different output encodings. Note that there are 4! = 24 different output encodings. In the figure below, the output encoding is chosen to minimize the array size. If the encoding for Sum were the same as that for Carry, the PLA would require nine columns.

Literal generator.

Four-Valued Signal	Two-Valued Signals			
0	0	3	3	3
1	3	0	3	3
2	3	3	0	3
3	3	3	3	0

Inverters with different thresholds.

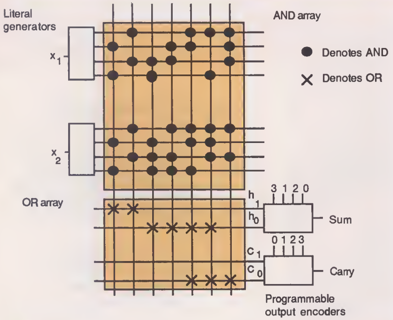
Input	Outputs ▷○	▷○	▷○	▷○
0	3	3	3	3
1	0	3	3	—
2	0	0	3	—
3	0	0	0	0

Output encoding for Sum

Four-Valued Signal	Two-Valued Signals	
0	3	3
1	0	3
2	3	0
3	0	0

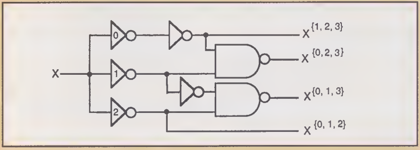
Output encoding for Carry.

Four-Valued Signal	Two-Valued Signals	
0	0	0
1	0	3
2	3	0
3	3	3

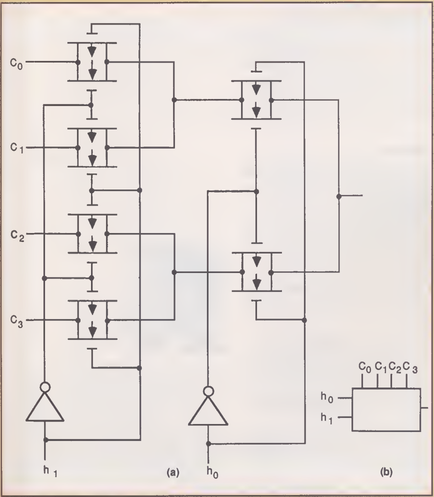


Four-valued PLA adder.

have a different number of inputs, and at least one decoder has more than two inputs. For adders, multipliers, and some control circuits, the array size of such PLAs is smaller than that of PLAs with two-bit decoders. In a PLA with multibit decoders, the number of literal lines is greater than that of a PLA with two-bit decoders. Thus, the design of such PLAs is more complex than that of PLAs with two-bit decoders. (See the sidebar "Example of PLAs with two-bit decoders.")



CMOS realization of a literal generator.



(a) CMOS realization of a programmable output encoder. (b) Logic symbol.

Multiple-output PLA

PLAs usually have multiple outputs. In designing multiple-output PLAs, independent minimization of each output does not always produce the full minimization. We have to consider all outputs at the same time. The literature of logical design often shows an extension of the Quine-McCluskey method, which uses multiple-output prime implicants. However, this method is complicated when treating the output part. A more elegant method is to use a characteristic function of the multiple-output function.[3] The characteristic function contains only and all the permitted combinations of inputs and outputs. In this function, we use an augmented variable X_{n+1} that shows the output part $(f_0, f_1, \ldots, f_{m-1})$ in addition to the input variables $x_1, x_2, \ldots, x_n$. The minimization algorithm for a characteristic function is more uniform and in many cases appears to be faster than the one that treats each output separately.[4,5] (See the sidebar "MVITVOF and its expressions.")

Output phase optimization

When realizing a multiple-output function by a PLA, we often have the option to realize f_i or $\bar{f}_i$ for each output. For each PLA output, we need a buffer either logically inverting or noninverting. We often can decrease the PLA's array size by appropriately selecting the phase (polarity) of each output.

For example, when the complement of f_0 is realized and an inverter is inserted into the output in Figure 6, we have a smaller PLA, as shown in Figure 7. In this case, the complementing of the first output reduces the array size.

The optimum output phase minimizes the total number of columns in the PLA. For an m-output function, there are 2^m possibilities of choosing the output phase. The exhaustive method for output phase optimization takes 2^m minimizations. This is impractical for large problems. I have developed a heuristic method that obtains good solutions in a reasonable computation time.[3] The heuristic algorithm uses the expressions for MVITVOFs. Table 2 compares the number of

PLA columns with and without consideration of the output phase. Intel used output phase optimization technique to reduce some of the PLAs used by its 80386 microprocessor.[10]

Encoding Problems

We can often reduce the size of PLAs by changing the encoding of the inputs or outputs.[4] Consider as an example of input encoding a PLA used for a processor's instruction decoder. In this case, we can choose the codes for the instruction to minimize the size of the PLA. As an example of output encoding, consider the generation of the control signals for a data path. In this case, we have to encode each action in the data path uniquely, but there is no reason to favor any particular encoding. Another example of output encoding is in a multiple-valued PLA.[11]

We can use MVITVOFs to solve the input encoding problem. An approximation to the state-assignment problem for finite-state machines can be solved by using MVITVOFs.[12] The output encoding problem is more difficult, but current techniques for solving the problem also use MVITVOFs. (See the sidebar "Multiple-valued PLAs and encoding problems.")

We can reduce industrial PLAs by 20 to 30 percent by using these new optimization techniques for multiple-output PLAs with decoders. For large PLAs, the use of two-bit decoders usually reduces the actual chip area even if we account for the decoder area.

Minimization of expressions for MVITVOFs is indispensable in these PLA optimizations. Modern microprocessors use PLAs extensively in their control parts, so high-quality minimizers for MVITVOFs, such as MINI-II and Espresso-MV, have become very important. Continuing efforts are being made to improve these optimization algorithms to work faster for larger PLAs.

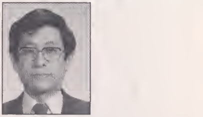
Logic minimization is a basic tool for VLSI logic design. Encoding problems, optimization of multilevel combinational networks, and decomposition of a large PLA into small linked PLAs are future applications of MVITVOFs. ☐

Acknowledgments

I wish to thank Jon T. Butler and all seven referees for refining this article, and S.D. Hamilton for reading the preliminary versions. This work is partially supported by a Grant in Aid for Scientific Research from the Ministry of Education, Science, and Culture of Japan, and a research grant from the Mazda Foundation.

References

1. H. Fleisher and L.I. Maissel, "An Introduction to Array Logic," *IBM J. Research and Development*, Vol. 19, No. 2, March 1976, pp. 98-109.
2. T. Sasao, "Multiple-Valued Decomposition of Generalized Boolean Functions and the Complexity of Programmable Logic Arrays," *IEEE Trans. Computers*, Sept. 1981, pp. 635-643.
3. T. Sasao, "Input Variable Assignment and Output Phase Optimization of PLAs," *IEEE Trans. Computers*, Oct. 1984, pp. 879-894.
4. R.L. Rudell and A. Sangiovanni-Vincentelli, "Multivalued Minimization for PLA Optimization," *IEEE Trans. CAD*, Sept. 1987, pp. 727-750, and also in *Proc. ISMVL-87*, May 1987, pp. 198-208.
5. R.K. Brayton et al., *Logic Minimization Algorithms for VLSI Synthesis*, Kluwer Academic Publishers, 1984.
6. P.L. Tison, "An Algebra for Logic Systems: Switching Circuit Application," *IEEE Trans. Computers*, Mar. 1971, pp. 339-351.
7. M. Davio, J.-P. Deschamps, and A. Thayse, *Discrete and Switching Functions*, Georgi Publishing Co., St.-Saphorin, Switzerland, 1978.
8. S.J. Hong, R.G. Cain, and D.L. Ostapko, "MINI: A Heuristic Approach for Logic Minimization," *IBM J. Research and Development*, Sept. 1974, pp. 443-458.
9. T. Sasao, "An Algorithm to Derive the Complement of a Binary Function With Multiple-Valued Inputs," *IEEE Trans. Computers*, Feb. 1985, pp. 131-140.
10. P.P. Gelsinger, "Design and Test of the 80386," *IEEE Design and Test*, June 1987, pp. 42-50.
11. T. Sasao, "On the Optimal Design of Multiple-Valued PLAs," *ISMVL-86*, May 1986, pp. 214-223, and also to be published in *IEEE Trans. Computers.*
12. G. De Micheli, R.K. Brayton, and A. Sangiovanni-Vincentelli, "Optimal State Assignment for Finite-State Machines," *IEEE Trans. CAD*, 1985, pp. 269-285.



Tsutomu Sasao is an associate professor of computer science and electronics at Kyushu Institute of Technology, Japan. His current interests include PLA design, automatic logic design systems, and complexity theory of logic circuits.

Sasao was the Asia Program Chairman of the 1984 International Symposium on Multiple-Valued Logic (ISMVL-84). He received the NIWA Memorial Award in 1979, and the Distinguished Contribution Award for ISMVL-86 in 1987.

He received his BS, MS, and PhD in electronic engineering from Osaka University in 1972, 1974, and 1977, respectively.

National Business Employment Weekly
THE WALL STREET JOURNAL

Jobs for computer pros!

Every week, the National Business Employment Weekly, published by The Wall Street Journal, contains hundreds of high-paying jobs from all across the country, including career opportunities in virtually every area of computer technology.

PLUS...weekly editorial features covering every aspect of career advancement.

"Computer" Issue.
April 24th.
Extra career opportunities.
for computer professionals.

Pick up the National Business Employment Weekly at your newsstand today. Or we'll send you the next eight issues by first class mail. Just send a check for $35 to:

National Business Employment Weekly
Dept. IC3, 420 Lexington Avenue
New York, NY 10170

Reader Service Number 6

ADVANCE ANNOUNCEMENT

CVPR'88

Conference on COMPUTER VISION AND PATTERN RECOGNITION

University of Michigan
Ann Arbor, MI
June 5-9, 1988

General Chair
Professor Ramesh Jain
Computer Science Department
University of Michigan
Ann Arbor, MI 48104

Program Chair
Professor Larry S. Davis
Center for Automation Research
University of Maryland
College Park, MD 20742

This year's conference will begin on Sunday, June 5, 1988 with half-day tutorial sessions dedicated to an intensive study of the following topics:

- Computer Vision For Autonomous Navigation
- Massively Parallel Models of Machine Intelligence
- Industrial Vision
- Range Image Processing

The technical sessions are divided into sessions that will include the presentation of papers in the following areas:

Monday, June 6, 1988
- Session 1.1—Calibration
- Session 1.2—Image Analysis
- Session 1.3—Stereo
- Session 1.4—Edge & Feature Extraction
- Session 1.5—Representation

Tuesday, June 7, 1988
- Session 2.1—Motion
- Session 2.2—Pattern Recognition
- Session 2.3—Shape Form
- Session 2.4—Image Processing and Applications
- Session 2.5—Geometry

Wednesday, June 8, 1988
- Session 3.1—Motion, 2
- Session 3.2—Morphology
- Session 3.3—Navigation
- Session 3.4—Matching/Recognition
- Session 3.5—Parallel Processing

The rates are as follows:

	Member	Non-member	Student/Retiree
Tutorial			
Early Registration (prior to May 1, 1988)	$ 75.00	$ 95.00	$ 75.00
Late/On-site Registration (after May 1, 1988)	$ 90.00	$115.00	$ 90.00
Conference			
Early Registration (prior to May 1, 1988)	$175.00	$220.00	$ 50.00
Late/On-site Registration (after May 1, 1988)	$200.00	$250.00	$ 50.00

Further Information:
Additional information, including the advance program and registration information, may be addressed to: The University of Michigan Extension Service, Department of Conferences and Institutes, 200 Hill Street, Ann Arbor, MI 48104-3297, (313) 764-5305.

THE COMPUTER SOCIETY OF THE IEEE

IEEE
THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC

11TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING

CHAIR
Larry Druffel
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890, USA
(412) 268-7740

PROGRAM CO-CHAIRS
Richard Fairley
School of Information Technology
George Mason University
Fairfax, VA 22030, USA
(703) 764-6195

Dines Bjørner
Dansk Datamatik Center
Lundtoftevej 1 C
Lyngby DK-2800, Denmark
(45) 2-872-622

TOOLS FAIR CHAIR
Grady Booch
Rational
835 South Moore Street
Lakewood, CO 80226, USA
(303) 986-2405

TUTORIALS CHAIR
Norman Gibbs
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890, USA
(412) 268-7703

LOCAL ARRANGEMENTS CHAIR
Miles Barel
Formative Technologies, Inc
Foster Plaza VII
661 Anderson Drive
Pittsburgh, PA 15220, USA
(412) 937-4942

To Be Sponsored By

acm ACM SIGSOFT

THE COMPUTER SOCIETY OF THE IEEE

Other sponsorships have been requested.

Call for Papers

Twenty Years Of Software Engineering: Looking Forward, Looking Back

Pittsburgh, Pennsylvania, USA May 15-18, 1989

The term "software engineering" was coined in 1968 to denote a new discipline concerned with the technological aspects of developing and modifying computer software systems. Twenty years have elapsed since the 1968-69 NATO Workshops on Software Engineering, which are generally acknowledged to be the seminal meetings. It is now appropriate to commemorate the past twenty years of software engineering and to look forward to the next twenty years.

In surveying the accomplishments of the past twenty years, the Conference will attempt to identify recurring themes and lasting contributions. In looking forward to the next twenty years, an attempt will be made to identify the major trends in hardware, software, and society that will shape the future of software engineering.

The Conference program will include both invited and contributed papers. Authors from all countries are encouraged to submit high-quality contributions in the mainstream topical areas of software engineering and in areas that are of considerable relevance to software engineering, such as database technology, expert systems, user interfaces, and hardware support for software engineering environments. Each submission will be evaluated by the Program Committee, who will assess the quality and relevance of the submitted papers. Proposals for panel sessions are also welcome. In addition, the Conference will feature a Software Tools Fair and a tools presentation track.

Submitting a Paper: Four copies (in English) should be submitted by September 15, 1988, to either Program Co-Chair:

- Richard Fairley
  School of Information Technology
  George Mason University
  Fairfax, VA 22030 USA
- Dines Bjørner
  Dansk Datamatik Center
  Lundtoftevej 1 C
  Lyngby DK-2800, Denmark

Papers should be limited to 6000 words, full-page figures being counted as 300 words. Each paper should include a short abstract and a list of keyworks indicating subject classification. Notification of acceptance will be sent by December 15, 1988, and camera-ready copy of the final version will be due on February 15, 1989.

Tools Fair: A Software Tools Fair will be held in parallel with the Conference. The Tools Fair will allow conference attendees to see and learn about current software tools. Both experimental and commercial tools will be demonstrated. In addition, the conference will feature a program track dedicated to presentation and demonstration of software tools selected by the Tools Fair Committee. Those interested in exhibiting a tool, and especially those authors interested in presenting a paper describing a particular tool and demonstrating that tool should contact the Tools Fair Chair.

- Grady Booch
  Rational
  835 South Moore Street
  Lakewood, CO 80226 USA

Further Information: For further information and/or a copy of the advance program when available, write to ICSE11, c/o IEEE Computer Society, 1730 Massachusetts Ave., N.W., Washington, D.C. 20036 USA.

Important Dates:

Submission Deadline: September 15, 1988

Acceptance Notification: December 15, 1988

Final Versions Due: February 15, 1989

IEEE THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Fuzzy Logic

Lofti A. Zadeh
University of California, Berkeley

Fuzzy logic — the logic underlying approximate, rather than exact, modes of reasoning — is finding applications that range from process control to medical diagnosis.

Logic, according to Webster's dictionary, is the science of the normative formal principles of reasoning. In this sense, fuzzy logic is concerned with the formal principles of approximate reasoning, with precise reasoning viewed as a limiting case.

In more specific terms, what is central about fuzzy logic is that, unlike classical logical systems, it aims at modeling the imprecise modes of reasoning that play an essential role in the remarkable human ability to make rational decisions in an environment of uncertainty and imprecision. This ability depends, in turn, on our ability to infer an approximate answer to a question based on a store of knowledge that is inexact, incomplete, or not totally reliable. For example:

(1) Usually it takes about an hour to drive from Berkeley to Stanford and about half an hour to drive from Stanford to San Jose. How long would it take to drive from Berkeley to San Jose via Stanford?

(2) Most of those who live in Belvedere have high incomes. It is probable that Mary lives in Belvedere. What can be said about Mary's income?

(3) Slimness is attractive. Carol is slim. Is Carol attractive?

(4) Brian is much taller than most of his close friends. How tall is Brian?

There are two main reasons why classical logical systems cannot cope with problems of this type. First, they do not provide a system for representing the meaning of propositions expressed in a natural language when the meaning is imprecise; and second, in those cases in which the meaning can be represented symbolically in a meaning representation language, for example, a semantic network or a conceptual-dependency graph, there is no mechanism for inference.

As will be seen, fuzzy logic addresses these problems in the following ways. First, the meaning of a lexically imprecise proposition is represented as an elastic constraint on a variable; and second, the answer to a query is deduced through a propagation of elastic constraints.

During the past several years, fuzzy logic has found numerous applications in fields ranging from finance to earthquake engineering. But what is striking is that its most important and visible application today is in a realm not anticipated when fuzzy logic was conceived, namely, the realm of fuzzy-logic-based process control. The basic idea underlying fuzzy logic control was suggested in notes published in 1968 and 1972[1,2] and described in greater detail in 1973.[3] The first implementation was pioneered by Mamdani and Assilian in 1974[4] in connection with the regulation of a steam engine. In the ensuing years, once the basic idea underlying fuzzy logic control became well understood, many applications followed. In Japan, in particular, the use of fuzzy logic in control processes is being pursued in many application areas, among them automatic train operation (Hitachi),[5] vehicle control (Sugeno Laboratory at Tokyo Institute of Technology),[5] robot control (Hirota Laboratory at Hosei University),[5] speech recognition (Ricoh),[5] universal controller (Fuji),[5] and stabilization control (Yamakawa Laboratory at Kumamoto University).[5] More about

0018-9162/88/0400-0083$01.00 © 1988 IEEE

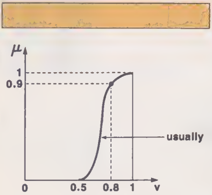


Figure 1. Representation of "usually" as a fuzzy proportion.

some of these projects will be said in the section dealing with applications.

In most of the current applications of fuzzy logic, software is employed as a medium for the implementation of fuzzy algorithms and control rules. What is clear, however, is that it would be cheaper and more effective to use fuzzy logic chips and, eventually, fuzzy computers. The first logic chip was developed by Togai and Watanabe at Bell Telephone Laboratories in 1985, and it is likely to become available for commercial use in 1988 or 1989. On the heels of this important development came the announcement of a fuzzy computer designed by Yamakawa at Kumamoto University. These developments on the hardware front may lead to an expanded use of fuzzy logic not only in industrial applications but, more generally, in knowledge-based systems in which the deduction of an answer to a query requires the inference machinery of fuzzy logic.

One important branch of fuzzy logic may be called *dispositional logic*. This logic, as its name implies, deals with *dispositions*, that is, propositions that are preponderantly but not necessarily always true. For example, "snow is white" is a disposition, as are the propositions "Swedes are blond" and "high quality is expensive." A disposition may be viewed as a usuality-qualified proposition in which the qualifying quantifier "usually" is implicit rather than explicit. In this sense, the disposition "snow is white" may be viewed as the result of suppressing the fuzzy quantifier "usually" in the usuality-qualified proposition

usually (snow is white)

In this proposition, "usually" plays the role of a fuzzy proportion of the form shown in Figure 1.

The importance of dispositional logic stems from the fact that most of what is usually referred to as common sense knowledge may be viewed as a collection of dispositions. Thus, the main concern of dispositional logic lies in the development of rules of inference from common sense knowledge.

In what follows, I present a condensed exposition of some basic ideas underlying fuzzy logic and describe some representative applications. More detailed information regarding fuzzy logic and its applications may be found in the cited literature.

Basic principles

Fuzzy logic may be viewed as an extension of multivalued logic. Its uses and objectives, however, are quite different. Thus, the fact that fuzzy logic deals with approximate rather than precise modes of reasoning implies that, in general, the chains of reasoning in fuzzy logic are short in length, and rigor does not play as important a role as it does in classical logical systems. In a nutshell, in fuzzy logic everything, including truth, is a matter of degree.

The greater expressive power of fuzzy logic derives from the fact that it contains as special cases not only the classical two-valued and multivalued logical systems but also probability theory and probabilistic logic. The main features of fuzzy logic that differentiate it from traditional logical systems are the following:

(1) In two-valued logical systems, a proposition p is either true or false. In multivalued logical systems, a proposition may be true or false or have an intermediate truth value, which may be an element of a finite or infinite truth value set T. In fuzzy logic, the truth values are allowed to range over the fuzzy subsets of T. For example, if T is the unit interval, then a truth value in fuzzy logic, for example, "very true," may be interpreted as a fuzzy subset of the unit interval. In this sense, a fuzzy truth value may be viewed as an imprecise characterization of a numerical truth value.

(2) The predicates in two-valued logic are constrained to be crisp in the sense that the denotation of a predicate must be a nonfuzzy subset of the universe of discourse. In fuzzy logic, the predicates may be crisp—for example, "mortal," "even," and "father of"—or, more generally, fuzzy—for example, "ill," "tired," "large," "tall," "much heavier," and "friend of."

(3) Two-valued as well as multivalued logics allow only two quantifiers: "all" and "some." By contrast, fuzzy logic allows, in addition, the use of fuzzy quantifiers exemplified by "most," "many," "several," "few," "much of," "frequently," "occasionally," "about ten," and so on. Such quantifiers may be interpreted as fuzzy numbers that provide an imprecise characterization of the cardinality of one or more fuzzy or nonfuzzy sets. In this perspective, a fuzzy quantifier may be viewed as a second-order fuzzy predicate. Based on this view, fuzzy quantifiers may be used to represent the meaning of propositions containing fuzzy probabilities and thereby make it possible to manipulate probabilities within fuzzy logic.

(4) Fuzzy logic provides a method for representing the meaning of both nonfuzzy and fuzzy predicate-modifiers exemplified by "not," "very," "more or less," "extremely," "slightly," "much," "a little," and so on. This, in turn, leads to a system for computing with *linguistic variables*,[3] that is, variables whose values are words or sentences in a natural or synthetic language. For example, "Age" is a linguistic variable when its values are assumed to be "young," "old," "very young," "not very old," and so forth. More about linguistic variables will be said at a later point.

(5) In two-valued logical systems, a proposition p may be qualified, principally by associating with p a truth value, "true" or "false"; a modal operator such as "possible" or "necessary"; and an intensional operator such as "know" or "believe." Fuzzy logic has three principal modes of qualification:

- *truth-qualification*, as in

  (Mary is young) is not quite true,

  in which the qualified proposition is (Mary is young) and the qualifying truth value is "not quite true";
- *probability-qualification*, as in

  (Mary is young) is unlikely,

  in which the qualifying fuzzy probability is "unlikely"; and
- *possibility-qualification*, as in

  (Mary is young) is almost impossible,

  in which the qualifying fuzzy possibility is "almost impossible."

An important issue in fuzzy logic relates to inference from qualified propositions, especially from probability-qualified propositions. This issue is of central importance in the management of uncertainty in expert systems and in the formalization of common sense reasoning. In the latter, it's important to note the close connection between probability-qualification and usuality-qualification and the role played by fuzzy quantifiers. For example, the disposition

Swedes are blond

may be interpreted as

most Swedes are blond;

or, equivalently, as

(Swede is blond) is likely,

where "likely" is a fuzzy probability that is numerically equal to the fuzzy quantifier "most"; or, equivalently, as

usually (a Swede is blond),

where "usually" qualifies the proposition "a Swede is blond."

As alluded earlier, inference from propositions of this type is a main concern of dispositional logic. More about this logic will be said at a later point.

Meaning representation and inference

A basic idea serving as a point of departure in fuzzy logic is that a proposition p in a natural or synthetic language may be viewed as a collection of elastic constraints, $C_1, \ldots, C_k$, which restrict the values of a collection of variables $X = (X_1, \ldots, X_n)$.[6] In general, the constraints as well as the variables they constrain are implicit rather than explicit in p. Viewed in this perspective, representation of the meaning of p is, in essence, a process by which the implicit constraints and variables in p are made explicit. In fuzzy logic, this is accomplished by representing p in the so-called *canonical form*

$$p \rightarrow X \text{ is } A$$

in which A is a fuzzy predicate or, equivalently, an n-ary fuzzy relation in U, where $U = U_1 \times U_2 \times \ldots \times U_n$, and U_i, $i = 1, \ldots, n$, is the domain of X_i. Representation of p in its canonical form requires, in general, the construction of an explanatory database and a test procedure that tests and aggregates the test scores associated with the elastic constraints $C_1, \ldots, C_k$.[6]

In more concrete terms, the canonical form of p implies that the possibility distribution[6] of X is equal to A—that is,

$$\Pi_X = A \tag{1}$$

which in turn implies that

$$\text{Poss}\{X = u\} = \mu_A(u),\ u \in U$$

where μ_A is the membership function of A and $\text{Poss}\{X = u\}$ is the possibility that X may take u as its value. Thus, when the meaning of p is represented in the form of Equation 1, it signifies that p induces a possibility distribution Π_X that is equal to A, with A playing the role of an elastic constraint on a variable X that is implicit in p. In effect, the possibility distribution of X, Π_X, is the set of possible values of X, with the understanding that possibility is a matter of degree. Viewed in this perspective, a proposition p constrains the possible values that X can take and thus defines its possibility distribution. This implies that the meaning of p is defined by (1) identifying the variable that is constrained and (2) characterizing the constraint to which the variable is subjected through its possibility distribution. Note that Equation 1 asserts that the possibility that X can take u as its value is numerically equal to the grade of membership, $\mu_A(u)$, of u in A.

As an illustration, consider the proposition

$$p \triangleq \text{John is tall}$$

in which the symbol $\triangleq$ should be read as "denotes" or "is equal to by definition." In this case, X = Height(John), A = TALL, and the canonical form of p reads

Height(John) is TALL

where the fuzzy relation TALL is in uppercase letters to underscore that it plays the role of a constraint in the canonical form. From the canonical form, it follows that

$$\text{Poss}\{\text{Height (John)} = u\} = \mu_{\text{TALL}}(u)$$

where μ_{TALL} is the membership function of TALL and $\mu_{\text{TALL}}(u)$ is the grade of membership of u in TALL or, equivalently, the degree to which a numerical height u satisfies the constraint induced by the relation TALL.

When p is a conditional proposition, its canonical form may be expressed as "Y is B if X is A," implying that p induces a conditional possibility distribution of Y given X, written as $\Pi_{(Y|X)}$. In fuzzy logic, $\Pi_{(Y|X)}$ may be defined in a variety of ways,[7] among which is a definition consistent with the definition of implication in Lakasiewicz's L_{Aleph_0} logic. In this case, the conditional possibility distribution function, $\pi_{(Y|X)}$, which defines $\Pi_{(Y|X)}$, may be expressed as

$$\begin{aligned} &\pi_{(Y|X)}(u, v) = \\ &1 \wedge (1 - \mu_A(u) + \mu_B(v)), \\ &\qquad u \in U,\ v \in V, \end{aligned} \tag{2}$$

where

$$\pi_{(Y|X)}(u, v) \triangleq \text{Poss}\{X = u,\ Y = v\}$$

μ_A and μ_B denote the membership functions of A and B, respectively; and $\wedge$ denotes the operator min.

When p is a quantified proposition of the form

$$p \triangleq Q\ A\text{'s are } B\text{'s}$$

for example,

$$p \triangleq \text{most tall men are not very fat}$$

where Q is a fuzzy quantifier and A and B are fuzzy predicates, the constrained variable, X, is the proportion of B's in A's, with Q representing an elastic constraint on X. More specifically, if U is a finite set $\{u_1, \ldots, u_m\}$, the proportion of B's in A's is defined as the *relative sigma-count*

$$\Sigma\text{Count}(B/A) = \frac{\sum_j \mu_A(u_j) \wedge \mu_B(u_j)}{\sum_j \mu_A(u_j)} \tag{3}$$

$$j = 1, \ldots, m$$

where $\mu_A(u_j)$ and $\mu_B(u_j)$ denote the grades of membership of u_j in A and B, respectively. Thus, expressed in its canonical form, Equation 3 may be written as

$\Sigma Count(B/A)$ *is* Q

which places in evidence the constrained variable, X, in p and the elastic constraint, Q, to which X is subjected. Note that X is the relative sigma-count of B in A.

The concept of a canonical form pro-

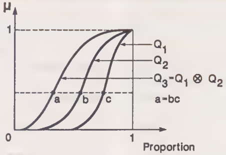


Figure 2. Representation of fuzzy quantifiers in the intersection/product syllogism.

vides an effective framework for formulating the problem of inference in expert systems. Specifically, consider a knowledge base, *KB*, which consists of a collection of propositions $\{p_1, \ldots, p_N\}$. Typically, a constituent proposition, p_i, $i = 1, \ldots, N$, may be (1) a fact that may be expressed in a canonical form as "*X* is *A*" or (2) a rule that may be expressed in a canonical form as "*Y* is B_i" if "*X* is A_i." More generally, both facts and rules may be probability-qualified or, equivalently, expressed as quantified propositions. For example, a rule of the general form "*Q A*'s are *B*'s" may be interpreted as the probability-qualified proposition (*X* is *B* if *X* is *A*) is λ, where λ is a fuzzy probability whose denotation as a fuzzy subset of the unit interval is the same as that of the fuzzy quantifier *Q* and *X* is chosen at random in *U*.

Now if p_i induces a possibility distribution $\Pi^i_{(X_1, \ldots, X_n)}$, where $X_1, \ldots, X_n$ are the variables constrained by p_i, then the possibility distribution $\Pi_{(X_1, \ldots, X_n)}$, which is induced by the totality of propositions in *KB* is given by the intersection[3] of the $\Pi^i_{(X_1, \ldots, X_n)}$. That is,

$$\Pi_{(X_1, \ldots, X_n)} = \Pi^1_{(X_1, \ldots, X_n)} \cap \ldots \cap \Pi^N_{(X_1, \ldots, X_n)}$$

or, equivalently,

$$\pi_{(X_1, \ldots, X_n)} = \pi^1_{(X_1, \ldots, X_n)} \wedge \ldots \wedge \pi^N_{(X_1, \ldots, X_n)}$$

$\pi_{(X_1, \ldots, X_n)}$ is the possibility distribution function of $\Pi_{(X_1, \ldots, X_n)}$. Note that there is no loss of generality in assuming that the constrained variables $X_1, \ldots, X_n$ are the same for all propositions in *KB* since the set $\{X_1, \ldots, X_n\}$ may be taken to be the union of the constrained variables for each proposition.

Now suppose that we are interested in inferring the value of a specified function $f(X_1, \ldots, X_n)$, $f\colon U \rightarrow V$, of the variables constrained by the knowledge base. Because of the incompleteness and imprecision of the information resident in *KB*, what we can deduce, in general, is not the value of $f(X_1, \ldots, X_n)$ but its possibility distribution, Π_f. By employing the extension principle,[8] it can be shown that the possibility distribution function of *f* is given by the solution of the nonlinear program

$$\pi_f(v) = \max_{u_1, \ldots, u_n} [\pi^1_{(X_1, \ldots, X_n)}(u_1, \ldots, u_n) \wedge \ldots \wedge \pi^N_{(X_1, \ldots, X_n)}(u_1, \ldots, u_n)] \quad (4)$$

subject to the constraint

$$v = f(u_1, \ldots, u_n)$$

where $u_j \in U_j$, $i = 1, \ldots, n$, and $v \in V$. The reduction to the solution of a nonlinear program constitutes the principal tool for inference in fuzzy logic.

Fuzzy syllogisms. A basic fuzzy syllogism in fuzzy logic that is of considerable relevance to the rules of combination of evidence in expert systems is the *intersection/product syllogism*—a syllogism that serves as a rule of inference for quantified propositions.[9] This syllogism may be expressed as the inference rule

$$\begin{array}{l} Q_1\ A\text{'s are } B\text{'s} \\ Q_2\ (A \text{ and } B)\text{'s are } C\text{'s} \\ \hline (Q_1 \otimes Q_2)\ A\text{'s are } (B \text{ and } C)\text{'s} \end{array} \quad (5)$$

in which Q_1 and Q_2 are fuzzy quantifiers, *A*, *B*, and *C* are fuzzy predicates, and $Q_1 \otimes Q_2$ is the product of the fuzzy numbers Q_1 and Q_2 in fuzzy arithmetic.[10] (See Figure 2). For example, as a special case of Equation 5, we may write

$$\begin{array}{l} \text{most students are single} \\ \text{a little more than a half of single} \\ \text{students are male} \\ \hline \text{(most} \otimes \text{a little more than a half) of} \\ \text{students are single and male} \end{array}$$

Since the intersection of *B* and *C* is contained in *C*, the following corollary of Equation 5 is its immediate consequence.

$$\begin{array}{l} Q_1\ A\text{'s are } B\text{'s} \\ Q_2\ (A \text{ and } B)\text{'s are } C\text{'s} \\ \hline \geq (Q_1 \otimes Q_2)\ A\text{'s are } C\text{'s} \end{array} \quad (6)$$

where the fuzzy number $\geq (Q_1 \otimes Q_2)$ should be read as "at least $(Q_1 \otimes Q_2)$." In particular, if the fuzzy quantifiers Q_1 and Q_2 are monotone increasing (for example, when "$Q_1 = Q_2 \triangleq$ most"), then

$$\geq (Q_1 \otimes Q_2) = Q_1 \otimes Q_2$$

and Equation 6 becomes

$$\begin{array}{l} Q_1\ A\text{'s are } B\text{'s} \\ Q_2\ (A \text{ and } B)\text{'s are } C\text{'s} \\ \hline (Q_1 \otimes Q_2)\ A\text{'s are } C\text{'s} \end{array} \quad (7)$$

Furthermore, if *B* is a subset of *A*, then *A* and $B = B$, and Equation 7 reduces to the *chaining rule*

$$\begin{array}{l} Q_1\ A\text{'s are } B\text{'s} \\ Q_2\ B\text{'s are } C\text{'s} \\ \hline (Q_1 \otimes Q_2)\ A\text{'s are } C\text{'s} \end{array} \quad (8)$$

For example,

$$\begin{array}{l} \text{most students are undergraduates} \\ \text{most undergraduates are young} \\ \hline \text{most}^2 \text{ students are young} \end{array}$$

where "most^2" represents the product of the fuzzy number "most" with itself (see Figure 3).

What is important to observe is that the chaining rule expressed by Equation 8 serves the same purpose as the chaining rules in Mycin, Prospector, and other probability-based expert systems. However, Equation 8 is formulated in terms of fuzzy quantifiers rather than numerical probabilities or certainty factors, and it is a logical consequence of the concept of a relative sigma-count in fuzzy logic. Furthermore, the chaining rule (Equation 8) is robust in the sense that if Q_1 and Q_2 are close to unity, so is their product $Q_1 \otimes Q_2$. More specifically, if Q_1 and Q_2 are expressed as

$$Q_1 = 1 \ominus \varepsilon_1$$
$$Q_2 = 1 \ominus \varepsilon_2$$

where ε_1 and ε_2 are small fuzzy numbers, then, to a first approximation, Q may be expressed as

$$Q = 1 \ominus \varepsilon_1 \ominus \varepsilon_2$$

An important issue concerns the general properties Q_1, Q_2, A, B, and C must have to ensure robustness. As shown above, the containment of B in A and the monotonicity of Q_1 and Q_2 are conditions for robustness in the case of the intersection/product syllogism.

Another basic syllogism is the *consequent conjunction syllogism*

$$\begin{array}{l} Q_1\ A\text{'s are } B\text{'s} \\ \underline{Q_2\ A\text{'s are } C\text{'s}} \\ Q\ A\text{'s are } (B \text{ and } C)\text{'s} \end{array} \qquad (9)$$

where

$$0 \ovee (Q_1 \oplus Q_2 \ominus 1) \leq Q \leq Q_1 \owedge Q_2$$

in which the operators $\owedge$, $\ovee$, $\oplus$, $\ominus$, and the inequality $\leq$ are the extensions of $\wedge$, $\vee$, $+$, $-$, and $\leq$, respectively, to fuzzy numbers.

The consequent conjunction syllogism plays the same role in fuzzy logic as the rule of combination of evidence for conjunctive hypotheses does in Mycin and Prospector.[11] However, whereas in Mycin and Prospector the qualifying probabilities and certainty factors are real numbers, in the consequent conjunction syllogism the fuzzy quantifiers are fuzzy numbers. As can be seen from the result expressed by Equation 9, the conclusion yielded by the

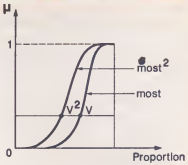


Figure 3. Representation of "most" and "most²."

Inference with fuzzy probabilities

An example of an important problem to which the reduction to a nonlinear program may be applied is the following. Assume that from a knowledge base $KB = \{p_1, \ldots, p_N\}$ in which the constituent propositions are true with probability one, we can infer a proposition q which, like the premises, is true with probability one. Now suppose that each p_i in KB is replaced with a probability-qualified proposition "$p_i^{\cdot} \triangleq p_i$ is λ_i," in which λ_i is a fuzzy probability. For example

$$p_i \triangleq X \text{ is small}$$

and

$$p_i^{\cdot} \triangleq X \text{ is small is very likely}$$

As a result of the qualification of the p_i, the conclusion, q, will also be a probability-qualified proposition that may be expressed as

$$q^{\cdot} = q \text{ is } \lambda$$

in which λ is a fuzzy probability. The problem is to determine λ as a function of the λ_i, if such a function exists. A special case of this problem, which is of particular relevance to the management of uncertainty in expert systems, is one in which the fuzzy probabilities λ_i are close to unity. We shall say that the inference process is *compositional* if λ can be expressed as a function of the λ_i; it is *robust* if whenever the λ_i are close to unity, so is λ.

By reducing the determination of λ to the solution of a nonlinear program, it can be shown that, in general, the inference process is not compositional if the λ_i and λ are numerical probabilities. This result calls into question the validity of the rules of combination of evidence in those expert systems in which the certainty factor of the conclusion is expressed as a function of the certainty factors of the premises. However, compositionality does hold, in general, if the λ_i and λ are assumed to be fuzzy probabilities, for this allows the probability of q to be interval-valued when the λ_i are numerical probabilities, which is consistent with known results in inductive logic.

Another important conclusion relating to the robustness of the inference process is that, in general, robustness does not hold without some restrictive assumptions on the premises. For example, the brittleness of the transitivity of implication is an instance of the lack of robustness when no assumptions are made regarding the fuzzy predicates A, B, and C. On the other hand, if in the inference schema

$$\begin{array}{l} X \text{ is } A \\ \underline{Y \text{ is } B \text{ if } X \text{ is } A} \\ Y \text{ is } B \end{array}$$

the major premise is replaced by "X is A is probable," where "probable" is a fuzzy probability close to unity, then it can be shown that, under mildly restrictive assumptions on A, the resulting conclusion may be expressed as "Y is B is $\geq$ probable," where "$\geq$ probable" is a fuzzy probability that, as a fuzzy number, is greater than or equal to the fuzzy number "probable." In this case, then, robustness does hold, for if "probable" is close to unity, so is "$\geq$ probable."

application of fuzzy logic to the premises in question is both robust and compositional.

A more complex problem is presented by what in Mycin and Prospector corresponds to the conjunctive combination of evidence. Stated in terms of quantified premises, the inference rule in question may be expressed as

$$\begin{array}{l} Q_1\ A\text{'s are } C\text{'s} \\ \underline{Q_2\ B\text{'s are } C\text{'s}} \\ Q\ (A \text{ and } B)\text{'s are } C\text{'s} \end{array} \tag{10}$$

where the value of Q is to be determined. To place in evidence the symmetry between Equation 9 and Equation 10, we shall refer to the rule in question as the *antecedent conjunction syllogism*.

It can readily be shown that, without any restrictive assumptions on Q_1, Q_2, A, B, and C, there is nothing that can be said about Q, which is equivalent to saying that "Q = none to all." A basic assumption in Mycin, Prospector, and related systems is that the items of evidence are conditionally independent, given the hypothesis (and its complement). That is,

$$P(E_1, E_2 \mid H) = P(E_1 \mid H)\, P(E_2 \mid H)$$

where $P(E_1, E_2 \mid H)$ is the joint probability of E_1 and E_2, given the hypothesis H; and $P(E_1 \mid H)$ and $P(E_2 \mid H)$ are the conditional probabilities of E_1 given H and E_2 given H, respectively. Expressed in terms of the relative sigma-counts, this assumption may be written as

$$\Sigma\text{Count}\,(A \cap B \,/\, C) = \Sigma\text{Count}\,(A \,/\, C)\, \Sigma\text{Count}\,(B \,/\, C) \tag{11}$$

where $\cap$ denotes the intersection of fuzzy sets.[3]

To determine the value of Q in Equation 10 we have to compute the relative sigma-count of C in $A \cap B$. It can be verified that, under the assumption (Equation 11), the sigma-count in question is given by

$$\Sigma\text{Count}\,(C \,/\, A \cap B) = \Sigma\text{Count}\,(C \,/\, A)\, \Sigma\text{Count}\,(C \,/\, B)\, \delta$$

where the factor δ is expressed by

$$\delta = \frac{\Sigma\text{Count}\,(A)\, \Sigma\text{Count}\,(B)}{\Sigma\text{Count}\,(A \cap B)\, \Sigma\text{Count}\,(C)} \tag{12}$$

Inspection of Equation 12 shows that the assumption expressed by Eqaution 11 does not ensure the compositionality of Q. However, it can be shown that compositionality can be achieved through the use of the concept of a relative *ϱsigma-count*, which is defined as

$$\varrho\Sigma\text{Count}\,(B/A) = \frac{\Sigma\text{Count}(B/A)}{\Sigma\text{Count}\,(\neg B/A)}$$

where $\neg B$ denotes the negation of B. The use of ϱsigma-counts in place of sigma-counts is analogous to the use of odds instead of probabilities in Prospector, and it serves the same purpose.

Interpolation

An important problem that arises in the operation of any rule-based system is the following. Suppose the user supplies a fact that, in its canonical form, may be expressed as "X is A," where A is a fuzzy or nonfuzzy predicate. Furthermore, suppose that there is no conditional rule in *KB* whose antecedent matches A exactly. The question arises: Which rules should be executed and how should their results be combined?

An approach to this problem, sketched in Reference 8, involves the use of an interpolation technique in fuzzy logic which requires a computation of the degree of partial match between the user-supplied fact and the rows of a decision table. More specifically, suppose that upon translation into their canonical forms, a group of propositions in *KB* may be expressed as a fuzzy relation of the form

R	X_1	X_2	.	X_n	X_{n+1}
	R_{11}	R_{12}	.	R_{1n}	Z_1
	.	.	.	.	.
	R_{m1}	R_{m2}	.	R_{mn}	Z_m

in which the entries are fuzzy sets; the input variables are $X_1, \ldots, X_n$, with domains $U_1, \ldots, U_n$; and the output variable is X_{n+1}, with domain U_{n+1}. The problem is: Given an input n-tuple $(R_1, \ldots, R_n)$, in which R_j, $j = 1, \ldots, n$, is a fuzzy subset of U_j, what is the value of X_{n+1} expressed as a fuzzy subset of U_{n+1}?

A possible approach to the problem is to compute for each pair (R_{ij}, R_j) the degree of consistency of the input R_j with the R_{ij} element of R, $i = 1, \ldots, m$, $j = 1, \ldots, n$. The degree of consistency, γ_{ij}, is defined as

$$\begin{aligned} \gamma_{ij} &\overset{\Delta}{=} \sup\,(R_{ij} \cap R_j) \\ &= \sup_{u_j}\,(\mu_{R_{ij}}(u_j) \wedge \mu_{R_j}(u_j)) \end{aligned}$$

in which $\mu_{R_{ij}}$ and μ_{R_j} are the membership functions of R_{ij} and R_j, respectively; u_j is a generic element of U_j; and the supremum is taken over u_j.

Next, we compute the overall degree of consistency, γ_i, of the input n-tuple $(R_1, \ldots, R_n)$ with the ith row of R, $i = 1, \ldots, m$, by employing $\wedge$ (min) as the aggregation operator. Thus,

$$\gamma_i = \gamma_{i1} \wedge \gamma_{i2} \wedge \ldots \wedge \gamma_{in}$$

which implies that γ_i may be interpreted as a conservative measure of agreement between the input n-tuple $(R_1, \ldots, R_n)$ and the ith-row n-tuple $(R_{i1}, \ldots, R_{in})$. Then, employing γ_i as a weighting coefficient, the desired expression for X_{n+1} may be written as a "linear" combination

$$X_{n+1} = \gamma_1 \wedge Z_1 + \ldots + \gamma_m \wedge Z_m$$

in which + denotes the union, and $\gamma_i \wedge z_i$ is a fuzzy set defined by

$$\mu_{\gamma_i \wedge Z_i}(u_{i+1}) = \gamma_i \wedge \mu_{Z_i}(u_{i+1}),\ i = 1, \ldots, m$$

The above approach ceases to be effective, however, when R is a sparse relation in the sense that no row of R has a high degree of consistency with the input n-tuple. For such cases, a more general interpolation technique has to be employed.

Basic rules of inference

One distinguishing characteristic of fuzzy logic is that premises and conclusions in an inference rule are generally expressed in canonical form. This representation places in evidence the fact that each premise is a constraint on a variable and that the conclusion is an induced constraint computed through a process of constraint propagation — a process that, in general, reduces to the solution of a nonlinear program. The following briefly presents — without derivation — some of the basic inference rules in fuzzy logic. Most of these rules can be deduced from the basic inference rule expressed by Equation 4.

The rules of inference in fuzzy logic may be classified in a variety of ways. One basic class is *categorical rules*, that is, rules that do not contain fuzzy quantifiers. A more general class is *dispositional rules*, rules in which one or more premises may contain, explicitly or implicitly, the fuzzy quantifier "usually." For example, the inference rule known as the *entailment principle*:

$$\begin{array}{l} X \text{ is } A \\ A \subset B \\ \hline X \text{ is } B \end{array} \tag{13}$$

where X is a variable taking values in a universe of discourse U, and A and B are fuzzy subsets of U, is a categorical rule. On the other hand, the *dispositional entailment principle* is an inference rule of the form

$$\begin{array}{l} \text{usually } (X \text{ is } A) \\ A \subset B \\ \hline \text{usually } (X \text{ is } B) \end{array} \tag{14}$$

In the limiting case where "usually" becomes "always," Equation 14 reduces to Equation 13.

In essence, the *entailment principle* asserts that from the proposition "X is A" we can always infer a less specific proposition "X is B." For example, from the proposition "Mary is young," which in its canonical form reads

Age(Mary) is YOUNG

where YOUNG is interpreted as a fuzzy set or, equivalently, as a fuzzy predicate, we can infer "Mary is not old," provided YOUNG is a subset of the complement of OLD. That is

$$\mu_{\text{YOUNG}}(u) \subset 1 - \mu_{\text{OLD}}(u),\ u \in [0, 100]$$

where μ_{YOUNG} and μ_{OLD} are, respectively, the membership functions of YOUNG and OLD, and the universe of discourse is the interval [0, 100].

Viewed in a different perspective, the entailment principle in fuzzy logic may be regarded as a generalization to fuzzy sets of the inheritance principle widely used in knowledge representation systems. More specifically, if the proposition "X is A" is interpreted as "X has property A," then the conclusion "X is B" may be interpreted as "X has property B," where B is any superset of A. In other words, X inherits property B if B is a superset of A.

Among other categorical rules that play a basic role in fuzzy logic are the following. In all of these rules, $X, Y, Z, \ldots$ are variables ranging over specified universes of discourse, and $A, B, C, \ldots$ are fuzzy predicates or, equivalently, fuzzy relations.

Conjunctive rule.

$$\begin{array}{l} X \text{ is } A \\ X \text{ is } B \\ \hline X \text{ is } A \cap B \end{array}$$

where $A \cap B$ is the intersection of A and B defined by

$$\mu_{A \cap B}(u) = \mu_A(u) \wedge \mu_B(u), \quad u \in U$$

Cartesian product.

$$\begin{array}{l} X \text{ is } A \\ Y \text{ is } B \\ \hline (X,\ Y) \text{ is } A \times B \end{array}$$

where (X, Y) is a binary variable and $A \times B$ is defined by

$$\mu_{A \times B}(u, v) = \mu_A(u) \wedge \mu_B(v), \quad u \in U,\ v \in V$$

Projection rule.

$$\begin{array}{l} (X,\ Y) \text{ is } R \\ \hline X \text{ is } {}_X R \end{array}$$

where ${}_X R$, the projection of the binary relation R on the domain of X, is defined by

$$\mu_{{}_X R}(u) = \sup_v \mu_R(u,v), \quad u \in U,\ v \in V$$

where $\mu_R(u,v)$ is the membership function of R and the supremum is taken over $v \in V$.

Compositional rule.

$$\begin{array}{l} X \text{ is } A \\ (X,\ Y) \text{ is } R \\ \hline Y \text{ is } A \circ R \end{array}$$

where $A \circ R$, the composition of the unary relation A with the binary relation R, is defined by

$$\mu_{A \circ R}(v) = \sup_u (\mu_A(u) \wedge \mu_R(u,v))$$

The compositional rule of inference may be viewed as a combination of the conjunctive and projection rules.

Generalized modus ponens.

$$\begin{array}{l} X \text{ is } A \\ Y \text{ is } C \text{ if } X \text{ is } B \\ \hline Y \text{ is } A \circ (\neg B \oplus C) \end{array}$$

where $\neg B$ denotes the negation of B and the bounded sum is defined by

$$\mu_{\neg B \oplus C}(u,v) = 1 \wedge (1 - \mu_B(u) + \mu_C(v))$$

An important feature of the generalized modus ponens, which is not possessed by the modus ponens in binary logical systems, is that the antecedent "X is B" need not be identical with the premise "X is A." It should be noted that the generalized modus ponens is related to the interpolation rule which was described earlier. An additional point that should be noted is that the generalized modus ponens may be regarded as an instance of the compositional rule of inference.

Dispositional modus ponens. In many applications involving common sense reasoning, the premises in the generalized modus ponens are usuality-qualified. In such cases, one may employ a dispositional version of the modus ponens. It may be expressed as

$$\begin{array}{l} \text{usually } (X \text{ is } A) \\ \text{usually } (Y \text{ is } B \text{ if } X \text{ is } A) \\ \hline \text{usually}^2 \ (Y \text{ is } B) \end{array}$$

where "usually^2" is the square of "usually" (see Figure 4). For simplicity, it's assumed that the premise "X is A" matches the antecedent in the conditional proposition; also, the conditional proposition is interpreted as the statement, "The

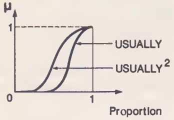


Figure 4. Representation of "usually" and "usually^2."

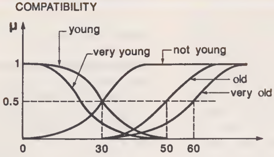


Figure 5. The linguistic values of "Age."

value of the fuzzy conditional probability of B given A is the fuzzy number USUALLY."

Extension principle. The extension principle plays an important role in fuzzy logic by providing a mechanism for computing induced constraints. More specifically, assume that a variable X taking values in a universe of discourse U is constrained by the proposition "X is A." Furthermore, assume that f is a mapping from U to V so that X is mapped into $f(X)$. The question is: What is the constraint on $f(X)$ which is induced by the constraint on X?

The answer provided by the extension principle may be expressed as the inference rule

$$\frac{X \text{ is } A}{f(X) \text{ is } f(A)}$$

where the membership function of $f(A)$ is defined by

$$\mu_{f(A)}(v) = \sup_u \mu_A(u) \qquad (15)$$

subject to the condition

$$v = f(u),\ u \in U,\ v \in V$$

In particular, if the function f is 1:1, then Equation 15 simplifies to

$$\mu_{f(A)}(v) = \mu_A(v^{-1}),\ v \in V$$

where v^{-1} is the inverse of v. For example,

$$\frac{X \text{ is small}}{X^2 \text{ is small}^2}$$

and

$$\mu_{\text{SMALL}^2}(v) = \mu_{\text{SMALL}}(\sqrt{v})$$

As in the case of the entailment rule, the dispositional version of the extension principle has the simple form

$$\frac{\text{usually } (X \text{ is } A)}{\text{usually } (f(X) \text{ is } f(A))}$$

The dispositional extension principle plays an important role in inference from common sense knowledge. In particular, it is one of the inference rules that play an essential role in answering the questions posed in the introduction.

The linguistic variable and its application to fuzzy control

A basic concept in fuzzy logic that plays a key role in many of its applications, especially in the realm of fuzzy control and fuzzy expert systems, is a *linguistic variable*.

A linguistic variable, as its name suggests, is a variable whose values are words or sentences in a natural or synthetic language. For example, "Age" is a linguistic variable if its values are "young," "not young," "very young," "old," "not old," "very old," and so on.

In general, the values of a linguistic variable can be generated from a *primary term* (for example, "young") its antonym ("old"), a collection of modifiers ("not," "very," "more or less," "quite," "not very," etc.), and the connectives "and" and "or." For example, one value of "Age" may be "not very young and not very old." Such values can be generated by a context-free grammar. Furthermore, each value of a linguistic variable represents a possibility distribution, as shown in Figure 5 for the variable "Age." These possibility distributions may be computed from the given possibility distributions of the primary term and its antonym through the use of attributed grammar techniques.

An interesting application of the linguistic variable is embodied in the fuzzy car conceived and designed by Sugeno of the Tokyo Institute of Technology.[5] The car's fuzzy-logic-based control system lets it move autonomously along a track with rectangular turns and park in a designated space (see Figure 6). An important feature is the car's ability to learn from examples.

The basic idea behind the Sugeno fuzzy car is the following. The controlled variable Y, which is the steering angle, is assumed to be a function of the state variables $X_1, X_2, X_3, \ldots, X_n$, which represent the distances of the car from the boundaries of the track at a corner (see Figure 7). These values are treated as linguistic variables, with the primary terms represented as triangular possibility distributions (see Figure 8).

The control policy is represented as a finite collection of rules of the form

R^i: if (X_1 is A^i_1) and . . . (X_n is A^i_n), then

$$Y^i = a^i_0 + a^i_1 X_1 + \ldots + a^i_n X_n$$

where R^i is the ith rule; A^i_j is a linguistic value of X_j in R^i; Y^i is the value of the control variable suggested by R^i; and $a^i_0, \ldots, a^i_n$ are adjustable parameters, which define Y^i as a linear combination of the state variables.

In a given state $(X_1, \ldots, X_n)$, the truth value of the antecedent of R^i may be expressed as

$$W^i = A^i_1(X_1) \wedge \ldots \wedge A^i_n(X_n)$$

where $A^i_j(X_j)$ is the grade of membership of X_j in A^i_j. The aggregated value of the controlled variable Y is computed as the normalized linear combination

$$Y = \frac{W_1 Y^1 + \ldots + W_n Y^n}{W_1 + \ldots + W_n} \qquad (16)$$

Thus, Equation 16 may be interpreted as the result of a weighted vote in which the value suggested by R^i is given the weight $W_i/(W_1 + \ldots + W_n)$.

The values of the coefficients $a^i_1, \ldots, a^i_n$ are determined through training. Training consists of an operator guiding a model car along the track a few times until an identification algorithm converges on parameter values consistent with the control rules. By its nature, the training process cannot guarantee that the identification algorithm will always converge on the correct values of the coefficients. The justification is pragmatic: the system works in most cases.

Variations on this idea are embodied in most of the fuzzy-logic-based control systems developed so far. Many of these systems have proven to be highly reliable and superior in performance to conventional systems.[5]

Since most rules in expert systems have fuzzy antecedents and consequents, expert systems provide potentially important applications for fuzzy logic. For example[11]:

IF the search "space" is moderately small
THEN exhaustive search is feasible

IF a piece of code is called frequently
THEN it is worth optimizing

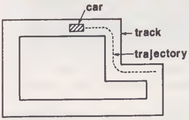


Figure 6. The Sugeno fuzzy car.

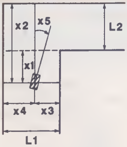


Figure 7. The state variables in Sugeno's car.

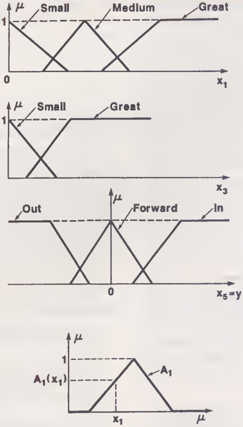


Figure 8. The linguistic values of state variables.

IF large oil spill or strong acid spill
THEN emergency is strongly suggested

The fuzziness of such rules is a consequence of the fact that a rule is a summary, and summaries, in general, are fuzzy. However, in the context of expert systems and fuzzy logic control, fuzziness has the positive effect of reducing the number of rules needed to approximately characterize a functional dependence between two or more variables.

Fuzzy hardware. Several expert system shells based on fuzzy logic are now commercially available, among them Reveal and Flops.[5] The seminal work of Togai and Watanabe at Bell Telephone Laboratories, which resulted in the development of a fuzzy logic chip, set the stage for using such chips in fuzzy-logic-based expert systems and, more generally, in rule-based systems not requiring a high degree of precision.[12] More recently, the fuzzy computer developed by Yamakawa of Kumamoto University has shown great promise as a general-purpose tool for processing linguistic data at high speed and with remarkable robustness.[5]

Togai and Watanabe's fuzzy inference chip consists of four major components: a rule set memory, an inference processor, a controller, and I/O circuitry. In a recent implementation, a rule set memory is realized by a random-access memory. In the inference processor, there are 16 data paths; one data path is laid out for each rule. All 16 rules on the chip are executed in parallel. The chip requires 64 clock cycles to produce an output. This translates to an execution speed of approximately 250,000 fuzzy logical inferences per second (FLIPS) at 16 megahertz clock. A fuzzy inference accelerator, which is a coprocessor board for a designated computer, is currently being designed. This board accommodates the new chips.

In the current implementation, the control variables are assumed to range over a finite set having no more than 31 elements. The membership function is quantized at 16 levels, with 15 representing full membership. Once the Togai/Watanabe chip becomes available commercially in 1988 or 1989, it should find many uses in both fuzzy-logic-based intelligent controllers and expert systems.

Yamakawa's fuzzy computer, whose hardware was built by OMRON Tateise Electronics Corporation, is capable of performing fuzzy inference at the very high speed of 10 megaFLIPS. Yamakawa's computer employs a parallel architecture. Basically, it has a fuzzy memory, a set of inference engines, a MAX block, and a defuzzifier. The computer is designed to process linguistic inputs, for example, "more or less small" and "very large," which are represented by analog voltages on data buses. A binary RAM, an array of registers and a membership function generator form the computer's fuzzy memory.

The linguistic inputs are fed to inference engines in parallel, with each rule yielding an output. The outputs are aggregated in the MAX block, yielding an overall fuzzy output that appears in the output data bus as a set of distributed analog voltages. In intelligent fuzzy control and other applications requiring nonfuzzy commands, the fuzzy output is fed to a defuzzifier for transformation into crisp output.

Yamakawa's fuzzy computer may be an important step toward a sixth-generation computer capable of processing common sense knowledge. This capability is a prerequisite to solving many AI problems — for example, handwritten text recognition, speech recognition, machine translation, summarization, and image understanding — that do not lend themselves to cost-effective solution within the bounds of conventional technology.☐

Acknowledgment

The research reported in this article has been supported in part by NASA Grant NCC-2-275 and NSF Grant DCR-8513139. The article was written while the author was a visiting scholar at the Center for the Study of Language and Information at Stanford University. It is dedicated to the Japanese scientists and engineers who have contributed so importantly to the development of fuzzy logic and its applications.

References

1. L.A. Zadeh, "Fuzzy Algorithms," *Information and Control*, Vol. 12, 1968, pp. 94-102.
2. L.A. Zadeh, "A Rationale for Fuzzy Control," *J.Dynamic Systems, Measurement and Control*, Vol. 94, Series G, 1972, pp. 3-4.
3. L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-3, 1973, pp. 28-44.
4. E.H. Mamdani and S. Assilian, "A Case Study on the Application of Fuzzy Set Theory to Automatic Control," *Proc. IFAC Stochastic Control Symp.*, Budapest, 1974.
5. *Preprints of the Second Congress of the International Fuzzy Systems Association*, Tokyo, Japan, 1987.
6. L.A. Zadeh, "Test-Score Semantics as a Basic for a Computational Approach to the Representation of Meaning," *Literary and Linguistic Computing*, Vol. 1, 1986, pp. 24-35.
7. D. Dubois and H. Prade, *Theorie des Possibilites*, Masson, Paris, 1985.
8. L.A. Zadeh, "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," *Fuzzy Sets and Systems*, Vol. 11, 1983, pp. 199-227.
9. L.A. Zadeh, "Syllogistic Reasoning in Fuzzy Logic and its Application to Usuality and Reasoning with Dispositions," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-15, 1985, pp. 754-763.
10. A. Kaufmann and M. M. Gupta, *Introduction to Fuzzy Arithmetic*, Van Nostrand, New York, 1985.
11. B. Buchanan and E. K. Shortliffe, *Rule-Based Expert Systems*, Addison-Wesley, Reading, Mass., 1984.
12. M. Togai and H. Watanabe, "Expert Systems on a Chip: An Engine for Real-Time Approximate Reasoning," *IEEE Expert*, Vol. 1, 1986, pp. 55-62.

Related Publications

Adams, E.W., and H.F. Levine, "On the Uncertainties Transmitted from Premises to Conclusions in Deductive Inferences," *Synthese*, Vol. 30, 1975, pp. 429-460.

Baldwin, J.F., and S.Q. Zhou, "A New Approach to Approximate Reasoning Using a Fuzzy Logic," *Fuzzy Sets and Systems*, Vol. 2, 1979, pp. 302-325.

Bellman, R.E., and L.A. Zadeh, "Local and Fuzzy Logics," in G. Epstein, ed., *Modern Uses of Multiple-Valued Logic*, Reidel, Dordrecht, 1977, pp. 103-165.

Dubois, D., and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.

Goguen, J.A., "The Logic of Inexact Concepts," *Synthese*, Vol 19, 1969, pp. 325-373.

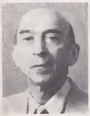
Klir, G.J., and T.A. Folger, *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall, Englewood Cliffs, N.J., 1988.

Moisil, G.C., *Lectures on the Logic of Fuzzy Reasoning*, Scientific Editions, Bucharest, 1975.

Prade, H., and C.V. Negoita, *Fuzzy Logic in Knowledge Engineering*, Verlag TÜV Rheinland, Köln, 1986.

Sugeno, M., ed., *Industrial Applications of Fuzzy Control*, Elsevier Science Publishers BV, The Netherlands, 1985.

Watanabe, H., and W. Dettloff, "Fuzzy Logic Inference Processor for Real Time Control: A Second Generation Full Custom Design," *Proc. 21st Asilomar Conference on Signals, Systems, and Computers*, Asilomar, Calif., 1987.



Lofti A. Zadeh joined the Department of Electrical Engineering at the University of California, Berkeley, in 1959, serving as its chairman from 1963 to 1968. Earlier he was a member of the electrical engineering faculty at Columbia University. He has held a number of visiting appointments, among them a visiting scientist appointment at the IBM Research Laboratory, San Jose, and a visiting scholar appointment at the AI Center, SRI International.

Zadeh's work centered on system theory and decision analysis until 1965 when his interests shifted to the theory of fuzzy sets and its applications.

An alumnus of the University of Teheran in Iran, MIT, and Columbia University, Zadeh is a fellow of the IEEE and AAAS and a member of the National Academy of Engineering. He holds a doctorate honoris causa from Paul Sabatier University, France, in recognition of his development of the theory of fuzzy sets.

Zadeh's address is Computer Science Division, Dept. of EECS, University of California, Berkeley, CA 94720.

SOFTWARE ENGINEERS

Tomorrow's Computing Technology is Today's Challenge

at

IDA

Some of the nation's most exciting developments in software technology, supercomputer architecture, AI, and expert systems are under scrutiny right now at the Institute for Defense Analyses. IDA is a Federally Funded Research and Development Center serving the Office of the Secretary of Defense, the Joint Chiefs of Staff, Defense Agencies, and other Federal sponsors.

IDA's Computer and Software Engineering Division (CSED) is seeking professional staff members with an in-depth theoretical and practical background in software engineering tools, techniques, and theories. One emphasis of the Division's work is on defining, building, and evaluating tools/environments to support the entire software life-cycle of mission critical computer based systems, focused on Ada and artificial intelligence environments. Tasks include efforts in prototyping tools/systems, testing and evaluating support tools, advising major DoD programs on appropriate technologies, and assisting the DoD in defining advanced policies for software development and maintenance.

Specific desired interests and skills include:

- **Formal representation of requirements and designs**
- **Program verification, test and evaluation**
- **Combined hardware/software system design methodologies**
- **Transformational approaches to automated program generation**
- **Software reusability and reuse**
- **Software reliability/fault tolerance**
- **Environment test and evaluation (including assessing impacts on productivity and reliability)**
- **Standards for environmental and operational software interfaces**

Specialists in other areas of Computer Science are also sought: **Distributed Systems, Artificial Intelligence and Expert Systems, Programming Language Experts and Computer Security Scientists.**

We offer career opportunities at many levels of experience. You may be a highly experienced individual able to lead IDA projects and programs . . . or a recent MS/PhD graduate. You can expect a competitive salary, excellent benefits, and a superior professional environment. Equally important, you can expect a role on the leading edge of the state of the art in computing. If this kind of future appeals to you, we urge you to investigate a career with IDA. Please forward your resume to:

Mr. Thomas J. Shirhall
Manager of Professional Staffing
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

An equal opportunity employer.
U.S. Citizenship is required.

IDA

Advance Program for the

15th Annual International Symposium on Computer Architecture

May 30th - June 3rd, 1988 - Ilikai Hotel, Honolulu, Hawaii, USA

Sponsored by
Computer Society of the Institute of Electrical and Electronics Engineers, TCCA
Association for Computing Machinery, SIGARCH

General Chair: H.J. Siegel, Supercomputing Research Center
Program Co-Chairs: Doug DeGroot, Texas Instruments
Yale N. Patt, UC, Berkeley

Symposium Committee:
Finance, Local Arrangements: Mary Jane Irwin, Penn State Univ.
Tutorials: Shreekant S. Thakkar, Sequent Computer Systems
Student Grants: Mark A. Franklin, Washington Univ.
Registration: Nathanial J. Davis IV, Air Force Inst. of Tech.
Publicity: Steven P. Levitan, Univ. of Pittsburgh
Proceedings: Jose Fortes, Purdue Univ.
Exhibits: Alan D. Berenbaum, AT&T Information Systems
Audio Visuals: S. Diane Smith, Tesuji

Program Committee:
Jean-Loup Baer, Univ. of Washington
Al Despain, UC, Berkeley
Jack Dongarra, Argonne Nat'l Labs
Mike Flynn, Stanford Univ.
Gideon Frieder, Syracuse Univ.
Dan Gajski, UC, Irvine
Lee Giles, Air Force Office of Scientific Research
Jim Goodman , Univ. of Wisconsin
Bert Halstead, MIT
Kai Hwang, Univ. of Southern California
Wen-mei Hwu, Univ. of Illinois
Joe Linn, Institute for Defense Analyses
Jack Lipovski, Univ. of Texas
Trevor Mudge, Univ. of Michigan
Janak Patel, Univ. of Illinois
Andy Pleszkun, Univ. of Wisconsin
Constantine Polychronopolous, Univ. of Illinois
Dan Siewiorek, Carnegie-Mellon Univ.
Dick Sites, Digital Equipment Corporation
Jack Stankovic, Univ. of Massachusetts

- 50 Outstanding Papers Selected from 275 Submissions
- Keynote Speaker David J. Kuck, CSRD, Univ. of Illinois
- Special Panels on Future Technologies and Future Parallel Computer Architectures
- 6 Half Day Tutorials and 1 Full Day Tutorial on State-of-the-Art Topics
- Eckert-Mauchly Award Ceremony

Talk to Colleagues at Tuesday's Cocktail Party, Wednesday's Banquet, and Thursday's (optional) Luau.

Dress at the conference is casual to reflect the relaxed atmosphere of Hawaii.

CONFERENCE SCHEDULE

ALL CONFERENCE ACTIVITIES WILL BE IN THE PACIFIC BALLROOMS

MONDAY, MAY 30, 1988:

8:00 - 9:00 a.m. Tutorial Registration, Pre-tutorial Coffee
9:00-10:00 a.m. Conference Registration

9:00 - 10:30 a.m. Morning Tutorials

(1) Functional Programming and Computer Architecture
Peter Harrison & Tony Field

(2) Parallel Processing Systems
H.J. Siegel

(3) GaAs Microprocessors: Architecture and Design
Veljko Milutinović

10:30 - 11:00 a.m. Tutorial Coffee Break

11:00 a.m. - 12:30 p.m. Morning Tutorials, con't.

12:30 - 2:30 p.m. Lunch Break (on your own)
1:30 - 2:30 p.m. Tutorial Registration

2:30 - 4:00 p.m. Afternoon Tutorials

(4) Logic Programming Architectures
John S. Conery & Evan Tick

(5) Fine Grained Concurrent Computing
William Dally & Andrew Chien

(6) Cache Design for Multiprocessor Systems
Shreekant S. Thakkar

4:00 - 4:30 p.m. Tutorial Coffee Break

4:30 - 6:00 p.m. Afternoon Tutorials, con't.

5:00 - 8:00 p.m. Conference Registration

TUESDAY, MAY 31, 1988

8:00 - 9:00 a.m. Pre-conference Coffee
8:00 a.m. - 2:00 p.m. Conference Registration

9:00 - 10:00 a.m. Session 1

Opening Session and Welcome

Keynote Address by David J. Kuck, CSRD, Univ. of Illinois

10:00 - 10:30 a.m. Conference Coffee Break

Session 2A: 10:30 a.m. - 12:00 noon
Neural Networks and Optical Computing
-*Critical Issues in Mapping Neural Networks on Message-Passing Multicomputers*, Joydeep Ghosh and Kai Hwang (Univ. of Southern California)
-*Multinomial Conjunctoid Statistical Learning Machines*, Yoshiyasu Takefuji, Robert Jannarone, Yong Cho and Tatung Chen (Univ. of South Carolina)
-*A Bit-Slice Array Architecture for Optical Computing with 2-D Symbolic Substitution*, Ahmed Louri and Kai Hwang (Univ. of Southern California)

Session 2B: 10:30 a.m. - 12:00 noon
Processor Design
-*The Reconfigurable Arithmetic Processor*, Stuart Fiske and William Dally (MIT)
-*The Performance Potential of Multiple Function Unit Processors*, Andrew Plezskun and Gurindar Sohi (Univ. of Wisconsin)
-*Exploiting Single-Chip Parallel Microarchitectures with a Compiler Code Generator: A Case Study*, Wen-mei Hwu and Pohua Chang (Univ. of Illinois)

12:00 noon - 2:00 p.m. Lunch Break (on your own)

Session 3A: 2:00 - 4:00 p.m.
Memory Hierarchies
-*Analysis of Memory Referencing Behavior for Design of Local Memories*, Geoffrey McNiven and Edward Davidson (Univ. of Illinois)
-*Performance Evaluation of On-Chip Register and Cache Organizations*, Richard Eickemeyer and Janak Patel (Univ. of Illinois)
-*On the Inclusion Properties for Multi-Level Cache Hierarchies*, Jean-Loup Baer and Wen-Hann Wang (Univ. of Washington)
-*A Simulation Study of Two-Level Caches*, Robert Short and Henry Levy (Univ. of Washington and DEC)

Session 3B: 2:00 - 4:00 p.m.
Networks - I
-*Hyperswitch Network for the Hypercube Computer*, E. Chow, H. Madan, J. Peterson (JPL), D. Grunwald, D. Reed (Univ. of Illinois)
-*Analysis of Bus Hierarchies for Multiprocessors*, Donald C. Winsor and Trevor N. Mudge (Univ. of Michigan)
-*Extra Group Network: A Cost-effective Fault-tolerant Multistage Interconnection Network*, Sizheng Wei and Gyungho Lee (Univ. of Southwestern Louisiana)
-*A Partial-Multiple-Bus Computer Structure with Improved Cost-Effectiveness*, Hong Jiang (Univ. of Southwestern Louisiana) and Kenneth C. Smith (Univ. of Toronto)

4:00 - 4:30 p.m. Conference Coffee Break

Session 4A: 4:30 - 6:00 p.m.
Functional/Dataflow Systems
-*Flagship: A Parallel Architecture for Declarative Programming*, Ian Watson, Viv Woods, Paul Watson, Richard Banach, Mark Greenberg, and John Sargeant (Univ. of Manchester, UK)
-*Toward a Dataflow/von Neumann Hybrid Architecture*, Robert Iannucci (MIT)
-*Resource Requirements of Dataflow Programs*, David Culler and Arvind (MIT)

Session 4B: 4:30 - 6:00 p.m.
Real Time Systems
-*Priority-Driven, Preemptive I/O Controllers for Real-Time Systems*, Brinkley Sprunt, David Kirk, and Lui Sha (Carnegie-Mellon Univ.)
-*A Kernel-Independent, Pipelined Architecture for Real-Time 2-D Convolution*, Shridhar B. Shukla and Dharma P. Agrawal (North Carolina State Univ.)
-*Exploiting Bit Level Concurrency in Real-Time Geometric Feature Extractions*, Wentai Liu and Tong-Fei Yeh (North Carolina State Univ.)

6:00 - 7:00 p.m. Conference Registration, Extra Tickets for Reception

7:00 - 9:00 p.m. POOL SIDE COCKTAIL RECEPTION

8:00 - 9:00 p.m. SIGARCH/TCCA BUSINESS MEETING

WEDNESDAY, JUNE 1, 1988

8:00 - 9:00 a.m. Conference Registration and Pre-conference Coffee

Session 5A: 9:00-10:30 a.m.
Characterization and Analysis
-*Measuring VAX 8800 Performance with a Histogram Hardware Monitor*, Douglas Clark, Pete Bannon, and Jim Keller (DEC)
-*Multiprocessor Address Tracing and Characterization Using ATUM*, Anant Agarwal (Stanford Univ.)
-*Trade-offs between Devices and Paths in Achieving Disk Interleaving*, Spenser Ng, Robert Selinger and Dorothy Lang (IBM)

Session 5B: 9:00-10:30 a.m.
Numeric Computation
-*Design of a Concurrent Computer for Solving Systems of Linear Equations*, K. Jainandunsing and E.F. Deprettere (Delft Univ. of Technology)
-*The White Dwarf: A High-Performance Application-Specific Processor*, A. Wolfe, M. Breternitz, C. Stephens, A. Ting, D. Kirk, R. Bianchini, and J. Shen (Carnegie-Mellon Univ.)
-*Solving Partial Differential Equations in a Data-Driven Multiprocessor Environment*, J.L. Gaudiot, M. Hosseiniyar and C.M. Lin (Univ. of Southern California)

10:30 - 11:00 a.m. Conference Coffee Break

Session 6A: 11:00 a.m. - 12:30 p.m.
Memory and Communication
-*A Scrambled Storage Scheme for Parallel Memory Systems*, De-lei Lee (York Univ.)
-*The Architecture of a Linda Coprocessor*, Venkatesh Krishnaswamy, Sudhir Ahuja, Nicholas Carriero, and David Gelernter (Yale Univ.)

Session 6B: 11:00 a.m. - 12:30 p.m.
Potpourri
-*Deadlock-Avoidance for Systolic Communication*, H.T. Kung (Carnegie-Mellon Univ.)
-*Cache Performance of Vector Processors*, Kimming So and Vittorio Zecca (IBM)
-*Distributed First-Come First-Served and Round-Robin Protocols and their Application to Multiprocessor Bus Arbitration*, Mary Vernon and Udi Manber (Univ. of Wisconsin)

WEDNESDAY, JUNE 1, 1988 (con't.)

12:30 - 2:30 p.m. Lunch Break (on your own)

Session 7A: 2:30 - 4:30 p.m.
Caches
-*Scalable Directory Schemes for Cache Coherence*, Anant Agarwal, Richard Simoni, John Hennessy, and Mark Horowitz (Stanford Univ.)
-*Performance Tradeoffs in Cache Design*, Steven Przybylski, Mark Horowitz and John Hennessy (Stanford Univ.)
-*Cache Coherence Scheme with Fast Selective Invalidation*, Hoichi Cheong and Alexander Veidenbaum (Univ. of Illinois)
-*An Accurate and Efficient Performance Analysis Technique for Multiprocessor Snooping Cache-Consistency Protocols*, Edward Lazowska, Mary Vernon, and John Zahorian (Univ. of Washington)

Session 7B: 2:30 - 4:30 p.m.
Networks - II
-*Destination Tag Routing Techniques Based on a State Model for the IADM Network*, Darwen Rau, Jose Fortes (Purdue Univ.) and H.J. Siegel (SRC)
-*Regular CC-Banyan Networks*, D.W. Kim, G.J. Lipovski (The Univ. of Texas), A. Hartmann (MCC) and Roy M. Jenevein (The Univ. of Texas)
-*High-Performance Multi-Queue Buffers for VLSI Communication Switches*, Yuval Tamir and Gregory Frazier (UCLA)

4:30 - 5:00 p.m. Conference Coffee Break

Session 8: 5:00 - 6:30 p.m.

Panel on Future Technologies

Chairman: C. Lee Giles (Air Force Office of Scientific Research). Panelists: Bidock Bedard (DOD) "Superconductivity," Sadik Esner (Dept. of EE & CS, UCSD, Call-Recall, Inc.) "3-D Memory," Alan Huang (AT&T Bell Labs) "Optical Computing Architectures," Larry Jackel (AT&T Bell Labs) "Electronic Neural Networks."

7:00 - 9:30 p.m. DINNER BANQUET
Eckert-Mauchly Award Ceremony

THURSDAY, JUNE 2, 1988

8:00 - 9:00 a.m. Conference Registration and Pre-conference Coffee

Session 9A: 9:00 - 10:30 a.m.
Multiprocessors-I
-*A Cache-based Message Passing Scheme for a Shared-bus Multiprocessor*, Bruno R. Preiss (Univ. of Waterloo) and V. Carl Hamacher (Univ. of Toronto)
-*IMPULSE: A High Performance Processing Unit for A Multiprocessor for Scientific Calculations*, Taisuke Boku, Shigehiro Nomura, and Hideharu Amano (Keio University)
-*Characterization of Sharing in Parallel Programs and Its Applicability to Coherency Protocol Evaluation*, Susan Eggers and Randy Katz (UC, Berkeley)

Session 9B: 9:00 - 10:30 a.m.
Synchronization Mechanisms
-*A Fetch-and-Op Implementation for Parallel Computers*, G.J. Lipovski and Paul Vaughan (The Univ. of Texas)
-*Synchronizing Processors through Memory Requests in a Tightly-coupled Multiprocessor*, Andre Seznec and Yvon Jegou (IRISA/INRIA)
-*Design and Performance of Special Purpose Hardware for Time Warp*, Richard M. Fujimoto, Jya Tsai, and Gnesh Gopalakrishnan (Univ. of Utah)

10:30 - 11:00 a.m. Conference Coffee Break

Session 10A: 11:00 a.m. - 12:30 p.m.
Multiprocessors-II
-*The VMP Multiprocessor: Initial Experience, Refinement and Performance Evaluation*, David Cheriton, Anoop Gupta, Patric Boyle, and Hendrick Goosen (Stanford Univ.)
-*The Wisconsin Multicube: A New Large Scale Cache-Coherent Multiprocessor*, James Goodman and Philip Woest (Univ. of Wisconsin)

Session 10B: 11:00 a.m. - 12:30 p.m.
AI Systems
-*Data Buffer Performance for Sequential Prolog Architectures*, Evan Tick (Stanford Univ.)
-*A Multithreaded Processor Architecture for Parallel Symbolic Computing*, Robert H. Halstead, Jr., (MIT) and Tetsuya Fujita (NEC)
-*Network Architecture for OPS5*, P.L. Butler, J.D. Allen, and D.W. Bouldin (Univ. of Tennessee)

12:30 - 2:00 p.m. Lunch Break (on your own)

Session 11: 2:00 - 3:30 p.m.

Panel on Future Directions in Parallel Computer Architecture

Chairman: Andre van Tilborg (Office of Naval Research). Panelists: Jack Dennis (MIT Lab for CS), H.T. Kung (Carnegie-Mellon Univ.), Burton Smith (Tera Computer), Harold Stone (IBM Research Center, Courant Inst.).

4:30 p.m. BUS FOR PARADISE COVE LUAU - Optional (IN FRONT OF THE ILIKAI)

FRIDAY, JUNE 3, 1988

8:00 - 9:00 a.m. Tutorial Registration and Pre-tutorial Coffee

9:00 - 10:30 a.m. Tutorial (7), Neural Networks, Clif Penn (Texas Instruments)

10:30 - 11:00 a.m. Tutorial Coffee Break

11:00 a.m. - 12:30 p.m. Tutorial, con't.

12:30 - 2:00 p.m. Tutorial Lunch (included)

2:00 - 3:30 p.m. Tutorial, con't.

3:30 - 4:00 p.m. Tutorial Coffee Break

4:00 - 5:30 p.m. Tutorial, con't.

15th Annual International Symposium on Computer Architecture

Monday (half day) and Friday (full day) Tutorials

Monday Morning: (half day tutorials)

(1) Functional Programming and Computer Architecture
Dr. Peter Harrison and Dr. Tony Field
Imperial College, London

In this tutorial we present an overview of functional languages and their application, and then consider the various ways in which they can be implemented. We shall look at the abstract computational models underlying these implementations and then describe some of the physical machines which have been built, or which have been proposed, based upon these models. These include: sequential *stack-based* machines, and distributed *graph reduction* machines, including *functional dataflow* machines. We shall relay some of our experiences with the ALICE machine, a prototype parallel graph reduction machine designed at Imperial College which has been operational since 1985. **Lecturer: Tony Field** (B.Sc., Reading, 1981, Ph.D., London (Imperial), 1985) is currently a lecturer in Computing at Imperial College. His interests include functional programming languages and implementations, parallel programming and interconnection networks. He is co-author of the book *Functional Programming*. **Lecturer: Peter Harrison** (B.A., M.A., Cambridge, 1972, 1976, M.Sc., Ph.D., London (Imperial), 1974, 1979) is currently a Lecturer in the Department of Computing at Imperial College. His current research interests include functional languages, their implementation and associated computer architectures, and mathematical models of performance. He is author of several papers in these areas and co-author of the book *Functional Programming*.

(2) Parallel Processing Systems
Dr. Howard Jay Siegel, Supercomputing Research Center

This tutorial examines the design of tightly-coupled large-scale parallel processing systems by presenting case studies of machines. MPP is an existing mesh connected SIMD 16k PE system. The Connection Machine is a 64k PE hypercube based SIMD machine. Ultracomputer is a MIMD system with a multi-stage cube fetch-and-add network (prototype under development). TRAC is a reconfigurable multi-stage network based SIMD/MIMD system (prototype built). PASM is a multi-stage cube network based partitionable SIMD/MIMD machine designed for image understanding (prototype built). **Lecturer: H.J. Siegel** (S.B. 1972 MIT, M.S., Ph.D., 1974, 1977 Princeton) is principal invetigator of the PASM research project at the Supercomputing Research Center. Previously, he was a Professor at Purdue. He is author of the book *Interconnection Networks for Large-Scale Parallel Processing*, and co-author of over 120 technical papers on parallel computer systems.

(3) GaAs Microprocessors: Architecture and Design
Dr. Veljko Milutinović, Purdue University

This tutorial explains several processor design strategies for GaAs technology, and emphasizes the RISC strategy. It discusses the impacts of GaAs technology on the design of CPU resources, system resources, and system software resources. It reviews a number of research projects in academia (Purdue, UCSB, RPI, CMU) and in industry, (DARPA, CDC, McDonnell Douglas, and RCA). Finally, experiments leading to the architecture and design of one 32-bit GaAs microprocessor are presented and analyzed. **Lecturer: Veljko Milutinović** is on the faculty of the School of Electrical Engineering, Purdue University. He has published about 30 journal papers, 2 original books, and 4 edited books. He served as the guest editor for the IEEE Computer *Special Issue on GaAs Computer Architecture and Related Topics*. His current interests include VLSI computer architecture for GaAs, technology-aware computer architecture, and computer architecture for signal processing.

Monday Afternoon: (half day tutorials)

(4) Logic Programming Architectures
Dr. John S. Conery, University of Oregon
Dr. Evan Tick, ICOT

The subject of this tutorial is architectural support for logic programming languages. We look at machine level operations in both sequential and parallel implementations of logic languages, such as Prolog and Concurrent Prolog, and survey current research projects. First, we present the Warren Abstract Machine (WAM). We discuss basic operations for unification, control flow, and memory usage. Next, we review the sources of parallelism in logic programming languages. We conclude with a look at several machines, such as the Berkeley PLM, the Gigalips system, and the KL-1 machine being developed at ICOT. **Lecturer: John Conery** (Ph.D., UC Irvine, 1983) is an Assistant Professor of Computer and Information Science at the University of Oregon. He is author of the book, *Parallel Execution of Logic Programs*. His current research project is the development of implementation techniques for OR-parallel systems on nonshared-memory multiprocessors. **Lecturer: Evan Tick** (Ph.D., Stanford, 1987) is currently a visiting researcher at the Institute for New Generation Computer Technology (ICOT) in Tokyo, Japan. He is author of the book, *Memory Performance of Prolog Architectures*. His current research project is the development of a detailed performance model for parallel logic programs.

(5) Fine Grained Concurrent Computing
Dr. William Dally and Andrew Chien, AI Lab. and LCS
Massachusetts Institute of Technology

In this tutorial, we describe the current state of the art in fine-grain concurrent computer systems: hardware and software. We begin with the presentation of two programming models for message-passing programming systems: a process model, and an actor model. With these two models in mind, we discuss the system services requirements. Memory management, task scheduling, code distribution and concurrency support issues are described. We address issues in the design of hardware for message passing systems. Key system services are identified and hardware support for these services is considered. Finally, we conclude with a discussion of remaining critical problems in fine-grain concurrent computer systems. **Lecturer: William J. Dally** (B.S., VPI, 1980, M.S. Stanford, 1981, Ph.D., Caltech, 1986) is currently an Assistant Professor of Computer Science at MIT. Bill worked at Bell Telephone Laboratories where he contributed to the design of the BELLMAC-32 microprocessor. Bill's research interests include concurrent computing, computer architecture, computer aided design, and VLSI design. **Lecturer: Andrew A. Chien** (S.B., S.M., MIT, 1984, 1987) is currently working towards his Ph.D. in MIT's Concurrent VLSI Architecture Group. His interests include architecture, networks, resource management, and performance evaluation.

(6) Cache Design for Multiprocessor Systems
Dr. Shreekant S. Thakkar, Sequent Computer Systems

This tutorial will cover cache coherence policies for both bus-based and network-based shared memory systems and reflect on their performance. The tutorial will address synchronization problems in shared memory multiprocessors. Bus protocols play a significant role in bus-based systems and they affect coherence schemes. A case study will show the role of these protocols in cache coherence scheme implementation. The tutorial will also address the use of virtual address caches, and multi-level caches in shared memory multiprocessors. **Lecturer: Shreekant Thakkar** (BSc., London (UCL), 1977, MSc., Ph.D., Manchester, 1978, 1982, C.Eng., 1986) is a senior systems engineer at Sequent Computer Systems responsible for the performance analysis of current and future system architectures. He has also been involved in the development of the Symmetry multiprocessor system. His research interests include highly parallel systems, computer architecture and performance, parallel programming and algorithms.

Friday, June 3: (full day tutorial)

(7) Neural Networks
Clif Penn, Texas Instruments

An introduction to neural nets at the tutorial level will be presented followed by a brief exposure to some of the experiments and applications which appear to have merit. Discussions of the methods used by modern practitioners will be covered for both supervised and unsupervised training of artificial neural networks. Some of the questions to be addressed are: What is a neural net? For what types of problems are they suited? Which methods work best for particular problems? Are special architectures beneficial? No prior knowledge of neural networks is required. **Lecturer: T.C. Penn** (B.S., Texas Tech., 1950, M.S., SMU, 1957) is a TI Fellow in the Computer Science Center at Texas Instruments. After joining TI 30 years ago, one of his first assignments was to design an artifical neuron using non-linear magnetics to implement the summation of the synaptic weights and neuron thresholds. Mr. Penn has numerous publications and patents. His team at TI has studied the use of neural networks for optimizing signal to noise ratios, text-to-speech, and speech recognition. He is the guest editor for the Feb. 1988 special issue of SIGARCH *Computer Architecture News* on Neural Networks.

15TH ISCA ADVANCE REGISTRATION FORM

FIRST NAME ______________________ LAST NAME ______________________

COMPANY/INSTITUTION ______________________

ADDRESS ______________________

CITY ______________________ STATE ______________ ZIP ____________

COUNTRY ______________ TELEPHONE ______________ E-MAIL ______________

FIRST NAME FOR BADGE ______________________

CHECK APPROPRIATE BOXES TO INDICATE MEMBER STATUS: ☐ACM ☐IEEE-CS ☐SIGARCH ☐TCCA

MEMBER NUMBER ______________________

FEES:	PRIOR TO MAY 8			AFTER MAY 8		
(Please Circle fee)	*STUDENT	MEMBER	NONMEMBER	*STUDENT	MEMBER	NONMEMBER
CONFERENCE REGISTRATION:	$40	$175	$220	$55	$205	$260
TUTORIALS: (Please circle selections and fees)						
#1, 2, OR 3	$50	$100	$150	$75	$125	$175
#4, 5, OR 6	50	100	150	75	125	175
#7 (includes lunch and textbook)	100	200	250	150	250	300

TOTAL CONFERENCE REGISTRATION FEE: $________

TOTAL TUTORIALS FEES: $________

ADDITIONAL COCKTAIL PARTY TICKETS: # ____ x $5 = $________

ADDITIONAL BANQUET TICKETS: # ____ x $35 = $________

CHECK HERE FOR VEGETARIAN BANQUET ☐

PARADISE COVE LUAU TICKETS: # ____ x $38 = $________

TOTAL FEE ENCLOSED US $________

ACM MAIL LIST RESTRICTION:

___ NO RESTRICTIONS

___ ACM/IEEE-CS AND OTHER SOCIETY ANNOUNCEMENTS

___ ACM/IEEE-CS ONLY

FOR OFFICE USE ONLY

Check # ________ Amount ________

Ribbon ______________

Date Entered ________ By ________

MAKE CHECKS PAYABLE IN US FUNDS ONLY TO: ACM/IEEE-CS 15TH ISCA

MAIL TO:
ACM/IEEE-CS 15TH ISCA
P.O. BOX 12105
CHURCH STREET STATION
NEW YORK, NY 10249

(Forms received after May 20 will not be processed.)

Full payment must accompany form or registration will not be processed.

Early registration discounts are extended to forms postmarked by May 8, 1988. ACM or IEEE-CS members must have membership number to receive discount. Cancellations must be made in writing and received by May 20, 1988. Conference registration includes access to all technical sessions, one copy of the proceedings, coffee breaks, Tuesday evening cocktail reception, and Wednesday evening banquet. Student registration includes access to all technical sessions, one copy of the proceedings, and coffee breaks. Proof of full time student enrollment must be included with the registration form. Additional banquet tickets and cocktail party tickets can be purchased for students and spouses. Proceedings will be available for sale onsite at the conference. Tutorial registration includes copies of notes, coffee breaks, and lunch for Friday's tutorial. *Scholarships for the tutorials are available for students attending them at their own expense (contact S. Thakkar (503) 627-9822). We reserve the right to cancel tutorials due to lack of registration.

Keep a copy of this form for your records.

HOTEL/CAR/TOUR RESERVATION FORM

Reservations must be returned to Irvine Travel along with a one night's deposit no later than thirty days prior to arrival date. Requests received after that date will be on a space available basis only. Deposit will be refunded if reservation is cancelled and notice is received at hotel 72 hours in advance of arrival date.

Irvine Travel Service is the official travel agency and only Irvine Travel can give conference discounts. Irvine Travel has contracted with Delta and United giving 5% off lowest applicable fare or 40% off coach. Call Irvine Travel for more information.

Send check and reservation form to:

IRVINE TRAVEL SERVICE, INC.
127 Northwestern Avenue
P.O. Box 2198
W. Lafayette, IN 47906
U.S. (except IN) 800-227-7477 or 317-743-2116

The Ilikai Waikiki -- Room Rates*
Tower Ocean View Rooms $100 single/double $115 triple
Tower Mountain View Rooms $95 single/double $110 triple
Yacht Harbor Ocean View Rooms $85 single/double $100 triple

Please check your hotel selection:
( ) single ( ) double ( ) triple
Please circle preferred category. If your selection is sold out, hotel will confirm the next available category.

Plus 9.41% Hawaii Sales Tax. Children 17 years and under sharing same room with parents are free.

*ALL HOTEL RESERVATIONS MUST BE MADE WITH IRVINE TRAVEL SERVICE

OPTIONAL TOURS

	# of Guests	$ Amount
City/Pearl Harbor/Moments in History Tour-Hosted - $21.50 pp	_____	_____
Night on the Town - $33.50 to $45.00 pp	_____	_____
Hanauma Bay Snorkel - $12.50 pp	_____	_____
Sea Life Park Tour - $25.50 pp	_____	_____
Moonlight Dinner Cruise - $35.00 pp	_____	_____
Polynesian Cultural Center Day Tour - $47.00 pp	_____	_____

CALL IRVINE TRAVEL SERVICE FOR DETAILS

Car Rental (Alamo, Budget, Hertz)
_____ Yes _____ No

Taxi Fare-airport to hotel-approx. $25. one way
Shuttle Service-airport to hotel $5.50 one way
(pickup outside baggage claim area)
Rental car parking at hotel - $6.75 per 24 hours

NAME ______________________

ADDRESS ______________________

CITY/STATE/ZIP ______________________

COUNTRY ______________ ARRIVAL DATE ______________ DEPARTURE DATE ______________

Check-out time is 3:00 p.m.

STANDARDS

Editor: Helen M. Wood, National Bureau of Standards, B154 Technology, Gaithersburg, MD 20899; (301) 975-3240

STEbus specification ratified as IEEE 1000

The IEEE Standards Board has approved the Standard Eurocard (STE) microcomputer bus architecture specification as IEEE 1000, establishing it as an internationally recognized standard.

The STEbus features a rugged eight-bit architecture, emphasizing processor and manufacturer independence as well as low cost. Most STEbus systems use a single-Eurocard form factor (100 mm × 160 mm), although IEEE 1000 also provides for the double Eurocard (233.35 mm × 160 mm) format.

"It's possible," according to IEEE 1000 coauthor Tim Elsmore of Wespercorp, "to see the STEbus as a kind of low-cost VMEbus that provides the reliability and flexibility of VME for users who don't need all of VME's performance." VME is the abbreviation for VersaModule European. "STE is also well-suited for use as an I/O channel with VMEbus or Futurebus," Elsmore added.

IEEE 1000 defines the STEbus as an architecture that may be used to "implement general purpose, high-performance, eight-bit microcomputer systems. Such systems may be used in stand-alone configurations or, in larger multiple bus architectures, as private (or secondary) buses, or high-speed I/O channels."

The authors of IEEE 1000 were careful to consider compatibility with the existing VMEbus standard, enabling the STEbus to form an I/O channel using VME's undefined row A and C P2 connector pin-outs.

About 30 vendors (including British Telecom) currently offer more than 700 STEbus board-level products that conform to the standard representing every major microprocessor family.

"STE's simplicity and elegance keep costs for these products low, contributing greatly to its emerging sales potential," said James Lyons of Val-Tech, chair of the STE Manufacturers and Users Group-US (STEMUG-US). "With design-ins and new systems, I expect the STEbus to capture a majority of the US embedded controller marketplace, as it has done already in the United Kingdom."

Typical STEbus applications include industrial process control, where STE's I/O capabilities have earned it widespread preference in the UK. Low-power, extended-temperature-range CMOS implementations are also possible, as exemplified by embedded control products developed by Val-Tech in the US. MS-DOS has also recently been moved onto the STEbus with ARCOM's STE PC System, thus providing full IBM PC compatibility.

Copies of IEEE 1000 may be obtained by contacting the Computer Society's Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-2578; (800) 272-6657 or (714) 821-8380.

For more information on STEbus architecture and STEbus-based products, contact STEMUG-US, PO Box 7529, Newark, DE 19714-7529, or STEMUG-UK, PO Box 149, Reading RG6 3HB, England, UK.

Government, industry establish digital-network users forum

Eight organizations have signed a charter forming the North American Integrated Services Digital Network Users Forum. Under an agreement signed February 3, the NIU-Forum will promote the development and implementation of standard interoperable products for the ISDN.

The founding organizations are Ameritech Services, the Association of Data Communications Users, Bell Atlantic, Bell Communications Research, International Computers, Ltd., the National Bureau of Standards, NYNEX, and Southwestern Bell Telephone.

ISDN is a new telecommunications technology that makes it possible to send and receive voice, data, and image signals simultaneously over digital telephone networks. Standards for ISDN are starting to emerge. However, because of the many options available in the standards, agreements on the specific options to implement are essential to achieve compatible products and services.

The NIU-Forum will work with established standards and testing organizations and the NBS's Workshop for Implementors of Open Systems Interconnection (OSI) to develop implementation agreements.

An ISDN Users Workshop will be formed under the NIU-Forum agreement to identify and discuss the requirements of industry and the private sector. The ISDN Implementors Workshop will define needed agreements, provide technical advice, and sponsor demonstrations and trials. Still another unit, the ISDN Conformance Testing Project, will be a permanent subgroup of the Implementors Workshop.

In addition to helping establish the forum, the NBS is developing standards and test methods for ISDN through its Institute for Computer Sciences and Technology. In conjunction with industry and other governmental organizations, the NBS has set up an ISDN laboratory.

Additional organizations are invited to join the ISDN users forum. For further information, contact Shukri Wakid at the NBS at (301) 975-2904.

For information on a particular ISDN workshop, contact: ISDN Users Workshop—Edward Hodgson, Association of Data Communications Users, (201) 397-6006; ISDN Implementors Workshop—Richard Stephenson, Southwestern Bell Telephone, (314) 235-1518; and ISDN Conformance Testing Project—Timon Holman, NYNEX, (914) 683-3033.

New standards publication issued for verifying, validating software

A new Federal Information Processing Standard Publication called *Guideline for Software Verification and Validation Plans*, has been issued for US government use in evaluating the quality of software.

The guideline, FIPS Pub 132, adopts ANSI/IEEE Standard 1012-1986, Standard for Software Verification and Validation Plans (SVVPs).

FIPS Pub 132 complements FIPS Pub 101, *Guideline for Lifecycle Validation, Verification, and Testing of Computer Software*, published in June 1983. These documents serve as tools for applying software verification and validation, a rigorous methodology for evaluating the accuracy and quality of software products during development and maintenance.

The US government currently spends $17 billion a year on computers and related equipment and services; this sum includes the acquisition and continued operation of many computer software applications. With FIPS Pub 132, those who acquire software will be able to monitor the quality of the progress for software development and maintenance to improve acceptance decisions at interim and final points of the acquisition process.

Software verification and validation (V&V) is a systems engineering methodology for evaluating quality and required performance. Software V&V requires careful planning for effective application of a rigorous methodology for evaluating the correctness and quality of the software product throughout the software life cycle.

The specifications for FIPS Pub 132 provide direction for preparing or assessing software verification and validation plans. The standard provides uniform and minimum requirements for the format and content of SVVPs.

The standard specifies minimum V&V tasks and the associated inputs and outputs that be included in SVVPs for critical software. It recommends their inclusion in SVVPs for noncritical software. The standard lists optional tasks to permit V&V planners to tailor their efforts to suit project needs.

ANSI/IEEE SVVP is part of a growing set of IEEE software engineering standards developed by voluntary working groups and approved by consensus ballots. The SVVP standard was developed over a 3½-year period by the Software Engineering Standards Subcommittee and was published in November 1986.

When approved as FIPS Pub 132 in November 1987, ANSI/IEEE Standard 1012-1986 became the first IEEE Software Engineering Standard to be adopted as a FIPS.

The National Bureau of Standards, through its Institute for Computer Sciences and Technology, worked with the Computer Society in developing the ANSI/IEEE standard and issued the new FIPS Pub. The NBS's objective is to help government agencies make optimum use of computers and related systems.

Copies of the new standard may be ordered from the Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-2578; (714) 821-8380.

The new FIPS, which includes ANSI/IEEE specifications, will soon be available to government agencies and the general public through the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

Computer security standard reaffirmed for five more years

The National Bureau of Standards has reaffirmed the use of the Data Encryption Standard for another five years, indicating the DES continues to be a sound and economical method for protecting unclassified but valuable or sensitive computerized data.

Based on a technique developed by IBM, the standard was adopted by the NBS and approved by the secretary of the US Department of Commerce as a Federal Information Processing Standard (FIPS) in 1977. (Such standards are developed by the NBS for use by the federal government.)

The DES is reviewed by the NBS and the commerce secretary every five years to determine its adequacy to protect computerized data. The standard is used widely in both the government and the private sector to protect data transmitted between computers and the data stored in them.

Among other uses, the DES has been adopted by the American National Standards Institute and the American Bankers Association to protect the transfer of funds and securities over communications lines.

FIPS Publication 46, DES, is available from the National Technical Information Service, Springfield, VA 22161, at a prepaid cost of $9.95. When ordering, specify FIPSPUB 46 and the title.

Ada/graphics draft ready for review

X3, the Accredited Standards Committee on Information Processing Systems, is conducting a four-month public review and comment period on draft proposed American National Standard, X3.144.3-198X. The review period continues until May 27.

This standard specifies the Ada language syntax for X3.144.3-198X, draft American National Standard for Information Systems—Computer Graphics—Programmer's Hierarchical Interactive Graphics System (PHIGS). The Ada language binding of PHIGS is a syntactic specification, presented as a set of procedures and/or functions that, taken as a whole, provide a means for accessing the semantics of PHIGS from an Ada application program.

The Ada procedure name, argument list, and argument data types are given from each PHIGS function. In addition, any special errors exclusively associated with the Ada language binding of PHIGS are specified and assigned unique-error numbers. The data type definitions used to define the Ada binding to PHIGS are defined in a separate package; the PHIGS procedures and/or functions are defined in a separate package called package PHIGS.

Copies of the draft standard may be obtained from Global Engineering Documents by dialing (800) 854-7179 in the US or (202) 429-2860 outside the US.

Standards Briefs

Working Group on Simulation (P1173) plans August meeting

The IEEE P1173 Working Group on Simulation will conduct its second meeting of 1988 on August 12 in Montreal, Quebec, Canada.

The group met March 17 during the 21st Simulation Symposium in Tampa, Fla., to discuss formalizing a discrete event simulation language and related issues.

The group's mandate is to define the functional components of a reliable discrete event simulation language in expectation of assisting a wide range of simulation-language developers and future environments.

The August session will include in-depth discussions of issues that arose at the March meeting. In addition, responsibilities will be delegated to members in the group.

Meetings are open to all interested. Those who'd like to participate or obtain further information should contact the group's chair, Oryal Tanir, at (514) 468-5524.

Standard proposed for local distributed data interface

Accredited Standards Committee X3 is conducting a four-month public review and comment period on draft-proposed American National Standard X3.167-198x. Public review will continue through May 1.

This standard specifies a network sub-layer protocol, a data-link layer protocol, a high-performance physical layer protocol, and a physical layer useful in computer configurations of up to 16 nodes in which no two nodes are separated by more than 90 meters.

This standard is designed for use at a data rate of 70 megabits per second; however, the protocol may be used at other data rates.

Copies of the draft documents may be obtained by calling Global Engineering Documents at (800) 854-7179 within the US or (714) 540-9870 outside the US.

P1149 standard draft being readied for balloting

The IEEE P1149 Testability Bus Standardization Committee, sponsored by the Test Technology Committee of the Computer Society, is preparing Draft 5 of its specification for initial balloting.

The proposed standard covers a complete testability approach that users may implement to whatever degree is appropriate to their applications as long as they adhere to the rules set forth in the specification document.

The specification lists four subsets which can be incorporated into a circuit design to make it more testable. The currently defined subsets are P1149.1 (Four-Line Minimum Serial Digital Subset); P1149.2 (Expanded Serial-Only Digital-Only Subset); P1149.3 (Real-Time Digital Subset); and P1149.4 (Real-Time Analog Subset Interface).

The Draft 5 specification includes descriptions of various testability implementation techniques, including VHSIC TM and ETM bus closest equivalents, as well as an appendix explaining the why and how of testability.

Copies of the draft may be obtained by contacting committee co-chair John Turino at Logical Solutions Technology, 310 W. Hamilton Ave., Suite 101, Campbell, CA 95008; (408) 374-3640.

Draft standard for C programming language reworked

The Accredited Standards Committee on Information Processing Systems (X3) is conducting its second public review and comment period on draft proposed American National Standard, X3.159-198X. The two-month review period will end April 12.

X3J11, the technical committee in charge of developing a standard for the C programming language released the new draft proposed standard for formal comment. It contains a number of changes based on comments received during the first review period.

During the prior period, the committee received 32 letters containing about 500 individual requests for changes. The comments the committee received pointed out the need to clarify or reword certain sections and led to committee discussions that resulted in either technical improvements to the draft standard and rationale or reaffirmation of previous decisions.

Major wording changes were made in an attempt to clarify the meaning of several sections and make them more precise. The types and type compatibility areas got particular attention.

The committee also received several requests from the international community for additional language support and changes. Since one of the main goals of the committee has been to develop a standard that would be acceptable as an International Standards Organization (ISO) standard as well as an American National Standards Institute (ANSI) standard, these requests were also carefully reviewed and considered. A significant change that resulted from this work was the addition of low-level support for multibyte characters such as those needed to represent Japanese and Chinese character sets.

Copies of the draft standard may be obtained by calling Global Engineering Documents at (800) 854-7179 in the US or (202) 429-2860 outside the US.

Moving?

PLEASE NOTIFY
US 4 WEEKS
IN ADVANCE

Name (Please Print)

New Address

City State/Country Zip

MAIL TO:
IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854

ATTACH
LABEL
HERE

- This notice of address change will apply to all IEEE publications to which you subscribe.
- List new address above.
- If you have a question about your subscription, place label here and clip this form to your letter.

Computer Society NEWS

Editor: Sallie Sheppard, Office of Associate Provost for Honors Program and Undergraduate Studies, Texas A&M University, College Station, TX 77843; (409) 845-3210

AT BOARD OF GOVERNORS MEETING

Editor-in-chief selected for *Transactions on Knowledge and Database Engineering*

Computer Society President Edward A. Parrish, Jr., has named C.V. Ramamoorthy as the first editor-in-chief of the new *IEEE Transactions on Knowledge and Data Engineering* that will start quarterly publication in 1989.

The presidential appointment was made on the recommendation of the society's Publications Board, chaired by Vice President James H. Aylor, and with the consent of the Board of Governors at its meetings in San Francisco March 4 during Compcon Spring 88.

Benjamin Wan-Sang Wah was selected to become the associate editor-in-chief.

The new transactions was approved by the society's Board of Governors last October and has received IEEE sanction.

Ramamoorthy of the University of California at Berkeley, Wah of the University of Illinois at Urbana-Champaign, and Bruce Berra of Syracuse University introduced the proposal for the new transactions last fall.

The journal's objective will be to provide an international forum for the publication of results on the research, design, and development of knowledge and data engineering methodologies, strategies, and systems.

It will be interdisciplinary and will cover such areas as computer science, artificial intelligence, electrical engineering, computer engineering, and cognitive science.

The focus will be on key technical issues related to the acquisition and management of data and knowledge about data in the development and utilization of information systems; the strategies to learn and store new knowledge acquired, reduce the burden of software and hardware development and maintenance, and provide improved overall functions and performance to meet new social needs; the mechanisms to provide system modeling and design, data access, and integrity control; the architectures, systems, and components to provide data services within centralized and distributed information systems and provide increased intelligence and ease of use through speech, voice, graphics, images, and documents; and the development of ways to prolong the useful life of data.

Ramamoorthy will select an editorial board soon.

Executive Committee appointments. The Computer Society by-laws call for the election of the president and first- and second-vice presidents and the appointment of the other officers by the president with the confirmation by the Board of Governors. President Parrish has appointed his slate of officers for 1988 and received the advice and consent of the Board of Governors at the March 4 meeting.

First Vice President Helen M. Wood will have the portfolio of the vice president for standards, and Second Vice President Joseph E. Urban will have the portfolio of the vice president for conferences and tutorials. The board also elected Duncan Lawrie as 1988 secretary.

The president's appointments of the other members of the executive committee are as follows:

- Vice president for area activities, Willis K. King,
- Vice president for education, Gerald L. Engel,
- Vice president for membership and information, Merlin G. Smith,
- Vice president for publications, James H. Aylor,
- Vice president for technical activities, Laurel V. Kaleda,
- Treasurer, Barry W. Johnson.

Immediate Past President Roy L. Russo, President-Elect Kenneth R. Anderson, Executive Director T. Michael Elliott, and the society's IEEE division directors, Troy Nagle and Harriett Rigas, also serve on the Executive Committee ex officio.

The next meeting of the Executive Committee is set April 15 in Brussels, Belgium, in conjunction with CompEuro 88. The full Board of Governors will meet June 16 in Anaheim, California, in conjunction with the 25th ACM/IEEE Design Automation Conference.

Computer Society Press report. Aylor, chair of the Ad Hoc Committee on the Future of the Computer Society Press, presented the final report of the committee calling for reorganization of the volunteer structure associated with CSP's activities.

A Computer Society Press Board is created under the plan. The board will be chaired by the vice president for Computer Society Press, a new position.

Establishing a new vice-president level position will increase the visibility of CS Press activities and facilitate attracting volunteers to serve in this area.

Currently, all CSP activities fall under the purview of the vice president for publications. The ad hoc committee suggested that that position is overloaded with consideration of both periodical and non-periodical publication issues.

In addition, the ad hoc committee considered activities that CSP should initiate and discussed the use of methods, such as electronic publishing, aimed at keeping the CS Press competitive in the future. Such products as abstracts of Computer Society conference proceedings, transactions, and magazines were considered.

The committee recommended increased quality and level of marketing effort for CS Press products. Because the budgets of libraries have not increased at the same rate as the costs of books and journals, many major university libraries are cutting back their subscriptions and book purchases. Increased marketing efforts, therefore, will be required just to maintain the current level of overall sales.

Top Computer Society awards presented at Compcon Spring 88

Ralph Preiss, 1987 Awards Chair

A number of top awards were presented at ceremonies conducted March 2 during Compcon Spring 88 in San Francisco, with Computer Society President Edward Parrish, Jr, doing the honors assisted by 1988 Awards Chair Claud Davis. Additional awards were presented during the ensuing luncheon for those award recipients who did not make it to the earlier ceremony.

The top 1987 technical award, the W. Wallace McDowell Award, established by IBM in 1966 to honor its vice president of engineering, went to Sidney Fernbach.

Four 1987 society-sponsored Computer Pioneer Awards were announced for Reynold B. Johnson, Arthur L. Samuel, Niklaus E. Wirth, and Robert R. Everett.

The 1987 Computer Entrepreneur Award, another society-sponsored presentation, went to Erwin Tomash.

The society also paid tribute to a number of volunteers and staff in the form of certificates.

In receiving the W. Wallace McDowell Award, Fernbach was recognized for his untiring work on supercomputing. Currently a consultant to Control Data Corporation, he spent most of his career as the director of the Lawrence Livermore National Laboratory computer department, which spawned most of the supercomputing requirements in the United States.

The four Computer Pioneer Award recipients were selected for making significant technical contributions to early concepts and developments that resulted in the creation and continued vitality of the electronic computer industry.

Johnson, a retired IBM engineer and manager, was honored for leading the team of engineers who developed the IBM 350 RAMAC, the computer industry's first commercial disk file.

Samuel, a retired IBM researcher, was singled out for developing adaptive non-numeric processing in the 1950s through his off-hours work on the checker-playing and learning program.

Wirth was honored for designing and developing the widely used language, Pascal, which is now an American National Standard.

A report on Compcon Spring 88 appears in the Conferences section of this issue of *Computer*.

Computer Society President Ed Parrish (top photo, left) presents Computer Pioneer Award to Niklaus Wirth during Compcon Spring 88 awards program. Also announced as recipients of the award were (not pictured) Reynold Johnson, Arthur Samuel, and Robert Everett. Sidney Fernbach (right photo) was presented the W. Wallace McDowell Award, and Erwin Tomash (bottom photo) received the Computer Entrepreneur Award.

Everett was cited for his work in developing the Whirlwind Project that led to the creation of the MIT Digital Computer Laboratory and later the MITRE Corporation, both of which he headed.

The society presents the Computer Entrepreneur Award to visionaries whose entrepreneurial efforts have resulted in the growth of some segment of the computer industry. Tomash fit the criteria of an entrepreneur from the outset of his career when, in 1952, he convinced Remington Rand management that the company should proceed with the ERA 1103 on a commercial basis (as a separate line from the Univac line).

Tomash saw the tremendous need for core memories in 1956 and made Telemeter Magnetics one of the leading manufacturers of that product. In 1962, he founded Dataproducts to expand into disc files and printers. He spearheaded Informatics into the software business and Data Card into the computer-driven magnetically encoded credit card embossing business. His exceptional ability to identify expanding markets for technological products and services was cited as earning him this recognition.

The following volunteer and staff service awards were announced and/or presented at Compcon:

Outstanding Contribution Certificates: Robert Blasewitz, Christina E. Champion, Richard Van Tilburg, and Eike Waltz.

Meritorious Service Certificates: Fletcher Buckley, Margaret Brown, Maris Graube, John Horch, Barry Johnson, Charles Radke, Ahmed Sameh, Sava Sherr, Earl Swartzlander, and Ron Waxman.

Certificates of Appreciation: Randy H. Katz, Martin Freeman, John Long, Dennis Reinhardt, and Lawrence Rosenblum.

Swartzlander was also presented the IEEE Fellow certificate, attesting to the fact that he had been elected to the IEEE Fellow grade.

Computer Society President Ed Parrish (left) formally presents IEEE Fellow Certificate to Earl Swartzlander at Compcon Spring 88 awards event. Swartzlander was one of 52 CS members elevated to fellow grade effective January 1, 1988 (see story on pp. 80-81 of the February 1988 issue of *Computer*).

Participants in the awards ceremonies at Compcon Spring 88 included (from left) Ron Waxman, Martin Freeman, Charles Radke, Niklaus Wirth, Barry Johnson, Christina Champion, Ed Parrish, John Horch, Earl Swartzlander, Margaret Brown, and Erwin Tomash.

NEWS FROM THE COMMITTEE ON PUBLIC POLICY

Does the United States need a national computer policy?

H. Taz Daughtrey COPP Newsletter Editor

No one would doubt either the present significance or the future potential of computing and associated technologies. Today, computer hardware, software, and related services account for up to 10 percent of the US Gross National Product. Positions of technological and economic leadership in the world rest increasingly on electronic products and the services they provide.

But, does the best hope for the US future lie in government-enunciated goals and government-coordinated projects? Perversely, the success of grand national policies often seems inversely proportional to the gravity of the situation and the surrounding rhetoric: witness waging the "War on Poverty" and seeking national energy independence as the "moral equivalent of war."

If Ben G. Matley and Thomas A. McDannold, the authors of the book, *National Computer Policies*, (Computer Society Press, 1987) do not declare war, they do at least sound a strong bugle call of warning. Matley and McDannold conclude that the American loss of computer markets is primarily due to the strategic plans of other nations.

Further, they claim that national positions in the future "information age" are being decided by present actions (or inactions) in formulating national computer policies.

What is a national computer policy (NCP)? Matley and McDannold define it as:

> ... a government-sponsored or endorsed document (or set of documents) that indicates the "official" role of the computer, and telecommunications in general, in terms of its relationship to the society and its business activities.

After a brief overview, the book is divided into three main sections: (1) reportage of established or emerging NCPs; (2) commentary, including proposals for a US national computer policy; and (3) wide-ranging pronouncements on the coming post-industrial "information age." The first two sections each cover 50-60 pages and are, in turn, informative and hortatory. The final, shorter section is presented as somewhat of an agenda of topics NCPs need to address for both the near- and longer-term future.

The first NCP considered is that of Japan. The MITI-sponsored 1972 study, *The Plan for Information Society—A National Goal Toward Year 2000*, presented an action plan that would be necessary for transition from a "smokestack industry" economy. Social, as well as economic, analysis were incorporated, addressing topics such as popular resistance to computerization and the need to promote intellectual creativity. Nine key projects were identified for the 1970s and early '80s.

France is next in the roll call of national computer policies, with *The Computerization of Society* (published in 1978). This vision of "telematics" encompassed satellite communications, digital telephone, and two-way television. As with the Japanese, the French based their study in part on social concerns such as the potential for cultural conflicts arising from decentralization and deindustrialization.

Great Britain's Alvey Committee published its NCP in 1982. This document is seen as the first of a series of product-specific programs that came to characterize subsequent NCPs. Rather than societal concerns, they would focus almost exclusively on economic issues such as market share and niches for specialized electronics products.

The 12-nation European Economic Community, whose ESPRIT (European Strategic Program for Research and Development in Information Technologies) project was launched in 1983 is covered next. This transnational NCP also focused on R&D for advanced microelectronics, software technology, and advanced information processing with commercial promise.

In the early 1980s, the Pacific Rim countries of South Korea, Taiwan, and Singapore also enunciated NCPs emphasizing domestic education and foreign trade initiatives.

The next tier of nations coming to grips with the economic demands of computerization has Brazil, Australia, and Israel. Each receive brief treatment on both its statement of policy and implementation through the mid-80s.

Interestingly, the book contains a full chapter on nations with no NCPs. The roll call reads: the USSR, West Germany, Sweden, Canada, the Netherlands, Ireland, China, and India.

Especially noteworthy was the observation that the closed Soviet society may never move beyond an industrial basis, not for lack of an NCP, but due to its very commitment to centralized control.

On the other hand, the very openness of Canada to transborder dataflows left it in the position of an undeveloped nation exporting low-value commodities (in this case, unprocessed data) and importing high-value finished products (information processed from the data).

At the end of this chapter, the authors claim, "While relatively few nations have formally adopted NCPs...all industrialized nations will eventually be drawn into NCP competition...."

Why has the US never developed a national computer policy? The authors attribute American indifference to two major factors: general aversion to government planning and what they call a 19th-century (pre-information) national mind-set. This is an interesting charge to level against as innovative and seemingly revolutionary a nation as the United States.

The Committee on Public Policy of the Computer Society has seen this book in the writing and highly recommends it as an introduction to thinking readers and people concerned with our apparent non-focused approach to the information age in which we find ourselves.

Where should we, the computer literate, ask our leaders to focus the nation's energies? As a nation, what expectations of technological leadership do we have? Can we maintain that leadership without a national computer policy? If we had such a policy, what forms would it take?

Readers interested in working on formulating a national computer policy should circle 188 on the reader service card in this issue of *Computer* and mail it in.

The COPP seeks and welcomes your professional opinions on policy issues.

The general public needs an informed view on computer-related topics, and it is up to the Computer Society, through its Committee on Public Policy, to provide the leadership for that informed view.

Society, ACM offering scholarships to design automation graduate students

The Computer Society's Technical Committee on Design Automation and its colleagues in the ACM SIGDA are sponsoring a scholarship program aimed at supporting graduate research and study in design automation.

DATC and SIGDA and have authorized the scholarship committee to award two new scholarships of $8,000 each. In addition, SIGDA has made up to $16,000 available for renewal of previous years' grants.

The scholarships are awarded directly to a university. Thus, the receiving university is free to use the funds in direct support of one or more of its design automation graduate students in the manner outlined in its proposal.

For publication in SIGDA's newsletter, the principal investigator of the project receiving a scholarship award is expected to submit a brief report of the year's activities that were supported by the scholarship.

Applications for new or renewed scholarships should be submitted by faculty advisors or department chairs and each should include: (a) a brief biography and transcript of the student(s) proposed for the scholarship; (b) a paragraph for each proposed student outlining his or her goals and objectives; (c) a brief proposal of the way in which the grant is to be expended; and (d) a brief statement of how the scholarship is expected to impact the institution's design automation program.

Applications are due no later than April 18, 1988 at the office of H. H. Loomis, Jr., Department of Electrical and Computer Engineering, Code 62Lm, Naval Postgraduate School, Monterey, CA 93943-5000.

Notification of the awards will be made in May.

For further information, contact Hersch Loomis at (408) 646-3214 or on MILNet at loomis@nps-cs.ARPA.

UPDATE

Contributions to Update are welcome. Send news of industrial or university research and of public policy or professional issues to Update Editor, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720, or to Bruce D. Shriver, Editor-in-Chief, IBM T.J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598.

National Science Foundation proposes $2.05 billion budget for fiscal 1989

Citing a need to stimulate research and education programs in light of growing international competition, the National Science Foundation is requesting a $2.05 billion budget appropriation for 1989, more than 19 percent above its 1988 budget of $1.72 billion.

The budget request, which is expected to reach the full House and Senate sometime this month, reflects a proposed rate of growth that would double the NSF budget over the next five years.

NSF's category for Computer and Information Science and Engineering would receive a 20.3 percent funding boost to $149.126 million in 1989 if the full budget is approved.

Within that general category, Networking and Communications Research and Infrastructure would receive $6.123 million, a 53 percent increase over 1988; Computer and Computation Research would receive $2.35 million, an 11.8 percent increase; Information Robotics and Intelligent Systems would receive $2.04 million, an 11.5 percent increase; Microelectronic Information Processing Systems would receive $2.765 million, a 20.9 percent increase; Advanced Scientific Computing would receive $11.465 million, a 25.9 percent increase; and Cross-Disciplinary Activities would receive $465,000, a 2.7 percent increase.

The foundation said it will continue to focus on three themes in 1989: education and human resources, disciplinary research programs and supporting facilities, and science and technology centers and groups.

Requested funding for disciplinary research and facilities is $1.2 billion—83 percent of the budget for research and related activities. Among the facilities to receive a substantial increase in financial support are the National Supercomputer Centers, which would benefit from a $11.3 million hike in funding to $64 million. Also, the National Science Foundation Network (NSFNET) would get a $5.9 million boost to $13.6 million.

Funding increases within disciplinary research are also sought in the areas of materials chemistry, including superconductivity, materials with designed properties, and processing of advanced materials for information storage and handling; manufacturing, including intelligent manufacturing systems, and novel engineering materials and processing technologies; biological communication, including studies aimed at neurosciences, human immunology, and plant defense systems; and cosmology, including attempts to discover more information about fundamental forces, the history of the universe, and possible new particles.

In education and human resources, programs for the kindergarten through 12th-grade level and the undergraduate level are targeted for increases of 21 percent (to $108 million) and 63 percent (to $65 million), respectively. Programs to improve participation in the sciences and engineering by minorities, women, disabled persons, and others would increase about 30 percent to $49 million under the proposed budget.

Plans in this area also include expansion of the Young Scholars Program from $3.7 million to $7 million; an increase in the number of initial-year graduate fellowships by 100, bringing the 1989 total to 860 new fellows; and continued support for Minority Research Initiation, Research Opportunities for Women, and Minority Centers of Research Excellence, which would receive $3.1 million, $2.9 million, and $5.8 million, respectively.

In its third targeted area, NSF has proposed that funding for university-based, multidisciplinary centers increase by $188 million, including a 10 percent expansion in existing center activities to $8 million, and a 17 percent increase to $30 million for group research.

Specific financial requests for group research activities include an increase from $1.5 million to $8.9 million for the Materials Research Group, which supports work in high-performance ceramics, plasticity and fracture of materials, and conducting polymers; and a provision of $3 million to establish mathematics research groups with an emphasis on integration across subfields and work with other disciplines.

The proposed budget for centers and groups also includes $150 million for initiating a new Science and Technology Research Centers program, which would feature upfront funding of 12 to 15 centers for periods up to five years.

The National Science Foundation's $1.72 billion budget for fiscal 1988 was reduced from a proposed appropriation of $1.89 billion. An NSF spokesman said attempts by Congress to adhere to balanced-budget legislation were partially responsible for the $170 million difference between the proposed and actual budgets. NSF expenditure in 1987 totalled $1.627 billion.

Nominations sought for Awards for Achievement

Nominations are being sought for the Awards for Achievement in Managing Information Technology, cosponsored by the Graduate School of Industrial Administration of Carnegie Mellon University and American Management Systems.

The awards recognize executives and professionals who have made outstanding contributions to their organizations through the effective use of computer systems and communications technology, according to the sponsors.

The 1988 awards will be presented at a dinner ceremony in New York in November. GSIA and AMS will accept submissions until May 1.

Finalists and winners will be selected by a panel of judges that will include AMS and other information executives, and GSIA faculty members. Each winner will receive a Steuben glass eagle and a cash honorarium.

For more information on award nominations, contact Mrs. Jan Dodson of AMS at (703) 841-5830.

MCC superconductivity program starts with 13

Thirteen companies will participate in the first phase of the Microelectronics and Computer Technology Corporation's research program to study the electronic applications of high-temperature superconductors. MCC is a cooperative research venture conducting research in microelectronics and computer technology.

The 13 participants are Bellcore, Boeing, Control Data, Digital Equipment, E.I. du Pont de Nemours, Eastman Kodak, General Electric, Harris, 3M, Motorola, NCR, Rockwell, and Westinghouse.

Work thus far has focused on modeling superconducting interconnects for computers and comparing their performance with that of chilled copper. Medium-thickness films of the superconductors have also been prepared.

Future research activities will include experiments, demonstrations of thin film preparation and patterning, and analysis of a hybrid (semiconductor-superconductor) system.

A company may participate in the program for $100,000. Membership in the MCC Associates Program costs $25,000 annually.

Results of the program are communicated to participants through reports and technical meetings. Companies can also send personnel to MCC on temporary assignment.

MCC management and participants will evaluate experimental and study results and determine priorities for follow-up efforts. If promising electronic applications are identified, phase two of the program will probably begin in 1989.

MCC is located at 3500 W. Balcones Center Dr., Austin, TX 78759-6509; (512) 343-0978.

Congressional Fellows choose their spots

IEEE's three 1987-1988 Congressional Fellows have chosen their assignments.

George C. Sponsler III is serving on the legislative staff of Sen. Paul Simon, studying issues and developing policy positions on the superconducting super collider, the Strategic Defense Initiative, a unified energy policy, civilian space policy, supercomputers, and acid rain.

James E. Gover chose to serve on the personal staff of Sen. Pete V. Domenici. He is responsible for technology issues and has already participated in activities involving semiconductors, competitiveness, and waste management.

Clark E. Johnson is serving on the personal staff of Rep. George Brown. He is working on competitiveness issues and on R&D funding as it relates to the House Science, Space, and Technology Committee.

The closing date for applications for the 1988-1989 Congressional Fellowships was March 31. Those who wish to participate in the program will need to apply for the 1989-1990 term.

For more information on the program, contact the IEEE Washington Office, 1111 19th St. NW, Suite 608, Washington, DC 20036; (202) 785-0017.

IDC holds 23rd annual information industry briefing session

International Data Corp. held its 23rd annual one-day briefing session in Washington, New York, Boston, San Jose, Calif., and Newport Beach, Calif., from March 8 to March 21. This year's topic was "Directions 88: An Industry in Transition."

J. Carl Masi, chair and CEO for IDS, predicted that 1988 will be a year of transition for the high-technology industry. Looking to the year 2000, he said that technology alone will not guarantee success, and that successful companies must master cyclical and structural change, understand technology integration issues, understand human resource issues, and implement market intelligence systems.

Masi felt that people issues will offer the major challenge in the next 12 years. Success or failure will depend partly upon the ability to anticipate a contracting workforce, computer literacy training, and changes in employee attitudes.

He also warned that a company run without a market intelligence system is doomed to failure.

William F. Zachman, senior VP of IDC, predicted booming growth in servers. He also predicted turnarounds for IBM's less enthusiastically supported products, such as AIX and the RT.

Other staff members of IDC presented detailed predictions of coming changes and trends.

For more information on the conclusions reached by this market research and analysis company, contact International Data Corp. in Framingham, Mass., at (617) 872-8200.

Foundation funds RIT research lab

The W.M. Keck Foundation of Los Angeles has awarded the College of Engineering at Rochester Institute of Technology $100,000 to enhance engineering research facilities in the Digital Image Processing Laboratory.

The grant is being given to RIT's Dept. of Electrical Engineering for high-speed computers and hardware to expand research and education in such areas as image enhancement, image restoration, and pattern recognition.

Edward Salem, director of the laboratory, anticipates research on the fundamentals of understanding images and how to manipulate them to extract usable information. He notes that image processing already helps doctors read X-rays with greater precision, assists scientists in interpreting images from space probes, and facilitates inventory control in manufacturing, construction of weather maps, and control of agricultural crops.

RIT is also in the process of constructing a Center for Imaging Science, scheduled for completion in 1989.

The W.M. Keck Foundation focuses on grants to universities and colleges in the United States, with an emphasis on the fields of science, engineering, and medicine.

NEW PRODUCT REVIEWS

Editor: Richard Eckhouse, MOCO, Inc., PO Box A, 91 Surfside Rd., Scituate, MA 02055; Compmail+, r.eckhouse

On buying a new 386-based PC

Richard Eckhouse, New Product Reviews editor

The pace of the personal computer marketplace continues to quicken. First, there is the constant barrage of new microprocessors, offering increased speed, expanded memory space, faster disks, and new video display standards. Second, there is the expanding supply of new vendors, each offering a faster machine at a lower price. As a result, I find myself wishing for something to replace my now ancient XT clone.

My introduction to PCs started with a floppy-disk-based 8088 machine running at 4.77 MHz. A year later I had an XT clone running at 8 MHz. Over time, I added another disk drive, an EGA display, and several minor items. Recently, I started thinking of upgrading to an AT, but the lure of going to a 20-MHz 30386-based machine with several megabytes of memory and a fast hard disk was enticing. Why not go to the top of the line and see what traveling in the fast lane was like, I reasoned. An AT would only be a temporary stopping point and the desire for a 386 would most likely remain.

I present all of this as the motivation for this review. Having talked with several colleagues, I got the feeling that others were in the same boat. The questions we all seemed to be asking were:

(1) Does it make sense to skip the 286-based machines and go on to those based on the 386?

(2) How much of a premium do you have to pay for a 386-based machine?

(3) Is the premium worth paying?

After looking at several new machines during my trip to Comdex, I carefully picked two vendors offering top-of-the-line machines. One was a 386 with a 16-MHz clock and no wait states, and the second a 20-MHz machine that had a remarkably low retail price. With retail prices for these machines not much more than an equivalent 286-based machine, I was convinced the answer to all my questions was to pay the premium because you do get noticeably faster CPU and disk performance. In fact, if you're willing to take a chance and buy an unknown clone, the price differential may be as little as $700 between a 386 running at 16 MHz and a 286 running at 10 MHz.

A number of features in addition to computation speeds set a 386 off from its cousins the 286 and the 86. First, most systems will come with a fast hard drive to match the speed of the 386 chip. The difference in disk performance is striking, particularly if you're moving up from an XT. Second, because of the 386 memory management capabilities, you will have more usable memory, much as you would on a 286 with a separate expanded-memory card. Third, there are a number of alternative operating environments or systems that you can now try. These systems include DESQview, Unix, PC-MOS/386, OS/2, and Windows/386, just to name a few.

With all these advantages, it might seem that there is little reason not to make your next computer a 386. However, I have not yet addressed one important issue. That issue comes under the heading of compatibility (or standardization).

The problem is that right now there really isn't a standard for designing and building a 386-based machine because there isn't the equivalent of the original IBM PC to clone. With IBM going in one direction in the form of the PS/2 series and the Micro Channel, and others, like Compaq, going the way of the AT but with a 386 microprocessor, there is really nothing to hang your hat on.

The real issue—compatibility

With no standard, what can you expect? For now, you will need to be content with a machine that runs MS- or PC-DOS, most other "enhanced" systems like DESQview, and some form of Unix. You may very well not be able to run OS/2 regardless of what the ads claim. The reason is that just using the 80386 as the microprocessor does not imply any level of compatibility. Depending on the system design and the BIOS level, the machine you purchase may or may not run certain programs and support certain plug-in boards. In many ways, it's just like the early days of the 8088 clones, when sellers proclaimed Lotus 1-2-3 and Flight Simulator compatibility. That was probably enough for the purchaser to falsely reason that the advertised machine would work, but many of us know that that was not always the case.

In fact, nothing much has changed from the early days when it comes to buying machines. As a prudent buyer, you must test the prospective new machine to verify that it is compatible with all the software and hardware you plan to use. That can be a little sticky when you don't own the items needed to test compatibility, they aren't available, or they cost too much to buy just to see if they work. Fortunately, most vendors are now offering a money-back guarantee with their systems. But what do you do when all you can get is assurances of future compatibility? Besides, do you really care about compatibility if you recognize that by the time you need it a new, faster, more compatible machine will have just been introduced?

I don't pretend to have the answers to all of these questions. I do know that of the several machines I've looked at, compatibility was definitely an issue. Fortunately, it was not an issue as far as the bus design went. All the machines opted for the AT bus design and they all ran at a nominal 8 MHz, so that every board I tested ran just as it would in an AT.

On the other hand, incompatibility in one machine meant that I couldn't use the Paradise VGA display adapter. In another, it meant that DESQview wasn't able to manage all the memory I had beyond the standard 640 kilobytes. And finally, I've been told by several of the system board manufacturers that their current 386-based machines do not run IBM's version of OS/2 because of its greater dependence on the actual hardware than DOS. Thus, until the Microsoft version of OS/2 is adapted to a specific vendor's machine, they will say little more than that their machines are currently "OS/2 ready."

To buy or not to buy

Even though compatibility is a sticky issue, I can say that I've come away with the feeling that a 386-based machine is a good buy. The caveat is that buying a 386-based machine shouldn't require a significant premium over a 286-based machine. Being on the cutting edge does present some problems that may take time to work out. However, the performance gain is well worth paying an extra $700 to $2,000. After all, some of the extra money goes into more memory or a faster disk. The price differential for the microprocessor and the motherboard is not that great.

On the subject of performance, one of the issues you will have to face is whether to buy what's offered today or wait for tomorrow's bigger and better system. I was made aware of this when I talked to several vendors. For instance, today's machine will either be slowed down to work with the 100-to-120-nanosecond memory chips, or will use an interleaved or static column RAM design to match memory to the processor speed. Indeed, if the processor is of the 20-MHz variety, the system designer will have to use some architectural technique to match memory to the processor cycle rate.

wait!

Whatever the solution, it is costly. There is bound to be something better just around the corner. Today, the cost of interleaved or static column RAM is considered too high, and the "new" solution is to use a cache (as in the 20-MHz 386-based machine from Compaq). But is the cache any better? It certainly saves money as far as having to buy the faster (and costlier) memory chips. However, it also can mean less performance because

(1) there is a penalty in filling the cache, repeated whenever the cache is flushed;

(2) the cache has to be designed to yield a high "hit rate," or it offers little advantage;

(3) the flush rate has to be low; and

(4) I/O-bound programs won't benefit from the cache and may not be cached at all.

So tomorrow's solution may be expedient and practical, but it may not be better.

Machines reviewed

My goal in looking at 386 machines from alternative vendors was to find out if they represented more performance at a lower cost. I was curious to know about the dollar saving plus the level of service and technical support I could expect. In all cases I was pleasantly surprised. I've commissioned a couple more reviews to appear later to provide additional insight.

While each of us reviewers had our own list of desirable features, we were able to pull them together in the form of an abbreviated checklist (see sidebar). In most cases, these features can make a sizeable difference in the bottom line of what you'll pay for a machine. They do not, however, tell you how well the machine will perform, whether or not it will work for its intended operation, and what the future will bring in the way of continued compatibility with new hardware and software. Compatibility is a difficult problem, one I've left up to each reviewer to describe with respect to the machine he chose to test.

The Heritage 386/16 from Computer Components Corp. operates at 16 MHz with zero wait states. It comes with the enhanced AT 101-key keyboard (not shown here).

Heritage 386/16

Right from the start you can see that a bit more effort has gone into the design of the Heritage 386/16 from Computer Components Corporation (or CCC). The stylish and solidly built enclosure features three accessible half-height drive bays. Because most users will fill only two of them, the third bay is not simply a flat panel, but a sculptured piece that fits the contour of the lower grill.

Most 386-based machines are a bit bigger than your average XT system, and this one is no exception. Externally, it measures approximately 21 inches across by 7 inches high by 17 inches deep. Inside, the spacious layout includes eight full-length and full-height slots, two of which accept 8-bit and six that accept 16-bit AT-style expansion boards. Actually, all of them could be 16-bit slots, since the vendor has laid out the motherboard so that AT-expansion bus connectors can be added at a later time.

With no 32-bit slots, you might wonder where the memory goes. This machine has been designed for up to 16 megabytes of memory using single in-line modules. With 256-kilobit chips, the first two megabytes are installed on the motherboard under the drive slots. Eight additional connectors positioned between the expansion board slots can hold up to two megabytes more. Later, when one-megabit chips are available, the four megabytes can be expanded to the grand total of 16 megabytes without requiring more space.

The motherboard, made by Tatung, is not crowded because of the excellent use of the latest VLSI chips from Chips and Technologies. One megabyte of memory is standard. The EGA board is from ATI Technologies, while the display is an enhanced version of NEC's well-known Multisync monitor. An I/O card features a serial/parallel port. The system also has a combined hard disk and floppy controller card. A time-of-day clock is part of the motherboard and it, along with the CMOS setup RAM, is powered by a single lithium cell. This combination leaves lots of free expansion slots.

The power supply, rated at 210 watts, can be switched between 115 and 230 volts. A power connector on the back for a monitor requires only the appropriate cable. The keyboard connector is also in the back, while the on/off switch is on the right side at the rear. The 101-key keyboard has a slightly nonstandard configuration.

The unit tested had a 40-megabyte full-height hard drive from Priam and one 1.2-megabyte floppy drive, with connectors for additional drives. The hard disk had an average seek time under 28 milliseconds and was well matched to the speed of the system. My evaluation unit also had a 10-MHz 80287 math coprocessor chip. Currently, the motherboard can only accept an 80287, but future machines will supposedly support the 80387 at a higher clock speed.

The lack of 387 support is one of the few things I did not like about this machine. This is admittedly a bit unfair, since most of my work doesn't require heavy coprocessor support; that which does uses it in bursts. But for those with heavy computational tasks, this might make a difference.

A second thing I didn't like was the feel of the keyboard. The touch was very heavy. Again, the plus or minus of this feature depends on the user—other people may not like the lighter style that I prefer.

My last complaint was not being able to set the processor clock speed to slow the machine down. While it's true that none of the software I used had problems at 16-MHz, a slow-down feature is a nice insurance policy to have.

To repeat, the system operates at 16 MHz with zero wait states. Adding a second megabyte allows you to interleave memory using the setup program and results in a significant increase in overall performance. The setup program also allows the Tatung BIOS to be relocated to RAM, which also provided a noticeable increase in execution speed.

Of all the machines I've seen, this one comes with the best documentation. While there is a minor problem with the translation into English, the user's manual is unusually complete. Details include both hardware and software descriptions, setting up the system, the power-on self-test, running the diagnostics, and what cables go where. Having taken the motherboard out of the cabinet, I was happy to have the manual to make sure that all connectors were replaced properly and that the setup RAM configured correctly. In fact, I often referred to this manual when I had questions about the other machine I was reviewing. This kind of help is darned hard to find, and Computer Components should be given high marks for standing out amongst the crowd.

An important part of the documentation is diagnostics. Once again the Heritage machine was outstanding—the user can run and fully test this machine at any time, and pinpoint a problem or at least have a standard against which to compare a malfunction. I consider this most important if you're buying mail-order or from a local vendor. In either case, software that helps to locate the problem in the system can greatly facilitate its speedy repair.

Another feature of the Heritage is that it comes with a one-year warranty with second-year and on-site service options available. This is very important for the more timid souls who don't intend to open the cabinet and who need repair service without the usual hassle of taking it back to the dealer. In my case, when I wanted technical support, a call to CCC always resulted in prompt and complete answers to all my questions.

I ran a full set of benchmarks on this machine. I measured millions of instructions per second, disk performance, memory read/write times, and floating-point performance. In all cases, I found the performance of the Heritage 386/16 significantly better than that of an equivalent 386 running at 16 MHz with zero wait states. For example, those upgrading from a 4.77-MHz PC or XT might expect to see about a nine-times performance increase, or more than twice that for an 8-MHz AT.

While you can buy this machine through CCC's distributors and dealers, you can also now go directly to the company. For $3,495 you get a one-megabyte monochrome graphics machine with disk controller, 40-megabyte hard

386 machine features

You will need to check for a number of features when buying a new personal computer, regardless of which model. I have listed those items that were particularly important to me when selecting the two machines reviewed in this issue of *Computer*.

- Type and speed of RAM (DRAM, static column, cache, etc.)
- How much memory on the motherboard?
- How much total memory?
- Coprocessor support (287 or 387)?
- What BIOS is used?
- Does it work with all peripherals and software?
- Clock speeds—selectable from keyboard?
- Bus speed—a standard 8 MHz?
- Slots available and filled?
- 32-bit slot for additional memory?
- Spaces for 5¼-inch or 3½-inch hard and floppy drives?
- Clock/calendar included?
- 40-to-80-megabyte hard disk available? What is the average seek time?
- 1.2-megabyte floppy drive included?
- MS-DOS included? What version?
- Utilities to manage memory or disk?
- Serial and parallel ports included and how many?
- EGA controller and display included? Multifrequency and what resolution?
- Type of keyboard?
- Reset switch?
- What status displays (power, speed, turbo mode, disk access, etc.)?
- Tape back-up—what capacity and cartridge type?

disk and 1.2-megabyte floppy, serial/parallel port, 101-key keyboard, and DOS 3.X and GW-Basic. A VGA system with multifrequency monitor is priced at $4,695. So, for a little more than half of what a comparable IBM or Compaq 386-based system might cost, you can have a machine with excellent performance and good national and on-site support.

Reader Service 20

The Darwin

One of the questions I wanted to answer when reviewing 386-based systems was, can you get a really good deal by buying an unknown clone from a local vendor? Having taken apart several other computers, I wasn't at all worried about opening the cabinet of a new machine. And since most clones contain a good many almost standard parts, such as the power supply, disk controller, serial card, and video adapter, I figured, how different could such a machine be?

I am located in the Boston area. I did my homework on local companies and chose PC Genius because its machines had received good reviews in other publications, it had sold machines to both university and corporate users, and it offered rock-bottom prices. After a few phone calls, I had a 20-MHz 386 with a megabyte of memory, a Seagate 40-megabyte hard disk, a 1.2-megabyte floppy, a Maxiswitch 101 keyboard, and a 20-MHz 387 coprocessor. The list price of the configuration I tested was $4,015. It included a monochrome monitor with a Hercules-compatible display adapter. Not bad for a top-of-the-line machine that goes by the name "the Darwin."

One of the first questions I had was, how good was the vendor's support? I found out fast when the Paradise VGA Plus display adapter I had planned to use failed to work. The machine wouldn't boot, and my only recourse was to call for help. The folks at PC Genius didn't have the answer, but they referred me to the manufacturer of the motherboard, Micronics Computers. What a pleasant group Micronics turned out to be. After years of not being able to deal directly with the manufacturer of my current systems, I was now able to talk with knowledgeable tech support people who patiently explained what was wrong and then rushed a new set of BIOS ROMs that completely solved the problem. Everything was up and running in less than two days.

The Darwin is housed in a plain but solid cabinet that measures 21 inches across by 7 inches high by 17 inches deep. Of the five half-height bays, three are externally accessible. Given that the hard disk and the floppy drive are half-height drives, you might expect to be able to add three more drives. However, the package includes only two more power and controller cables, indicating that four is the maximum number of devices the Western Digital controller board can handle.

There are eight expansion slots, one of which is used by the 80-nanosecond, static-column RAM memory board. This 32-bit board fits into a 32-bit bus that has its edge connector at the opposite end of the chassis from the other seven slots. Up to two megabytes fit on the memory board, and another two can be added using a daughter board. In both cases, the first megabyte is soldered onto the boards; additional sockets let you add another megabyte. At some point, these 256-kilobit chips can be replaced by one-megabit parts so that a full 16 megabytes can be configured.

Three other slots are taken up by the combined disk controller, the I/O card, and the display adapter. Normally, the system comes with two parallel and two serial ports, with one of the parallel ports emanating from the display adapter board. A feature that I liked, but others may not, is the use of standard alkaline AA cells to power the CMOS setup RAM and clock. To me, the savings over the cost of lithium batteries justifies the inconvenience of having to reset the system clock every time you replace the batteries.

The keyboard connector is in back and the power on/off switch at the rear on the right side. There are several cutouts for 25-pin D-type connectors, but none of them are used. A nice feature of the system is that it comes with a 9-pin to 25-pin adapter in case your modem or mouse isn't up to the AT standard for serial connectors.

MS-DOS and Basic are extra, but you do get a disk manager, a disk for setting up the Everex I/O board, and a number of utilities from Micronics. The disk manager takes care of testing and formatting the hard drive, and it also logically divides the drive into two smaller devices that MS-DOS can manage.

One of the utilities from Micronics was a setup program like that found on most ATs. Unfortunately, the setup program did not support the drive type supplied with the system. But PC Genius provided another one that did. A second program from Micronics is used to relocate the system and the EGA BIOS.

Another program allows you to adjust the system clock to one of five speeds, ranging from a slow XT through a 6-MHz or 8-MHz AT, up to the full-rated 20 MHz. A handy utility in the form of a memory management system allows the user to remap both the last 384 kilobytes of the first megabyte as well as the rest of expanded memory (everything above the first megabyte). I used this memory management software as a substitute for Quarterdeck's QEMM to work around the memory problem I report in that review.

As a check, I ran the usual set of benchmarks on the Darwin to see how it fared. The machine is faster in all respects than an equivalent 386 running at 16 MHz. The Landmark test showed a speed increase of 25 over a 4.77-MHz PC or XT. Hard-disk seek times were consistently under 30 milliseconds.

Next, I moved all of the programs I use every day onto the Darwin and proceeded to use them just as I had on my old and reliable Compaq Deskpro. Nearly everything ran, and it all ran noticeably faster. The one exception was Multiplan, and a call to Microsoft quickly solved the problem (as described in the following section, "A striking new display adapter"). I continued to use the system for over three weeks without a hitch, then called back to ask for an additional megabyte of memory. A quick exchange of boards and the staff at PC Genius had my system upgraded.

I have been using the Darwin 10 to 12 hours a day, and it continues to run smoothly and reliably. The 200-watt power supply runs cool, although somewhat more loudly than other systems. I haven't found any software—with the exception of QEMM—that isn't fully functional on the machine. All software runs when the 386 is operating at its full-rated speed.

Probably the least satisfying part of the clone experience is the documentation. What there is was supplied by the board manufacturer. Despite the wealth of information in the skinny Micronics booklet, it is too brief. I have to commend them for documenting their board as well as they have, even if the result is a bit skimpy. The Everex manual is more than sufficient for the I/O board. Finally, the Ontrack Computer Systems Disk Manager and Diagnostics for the Seagate hard disk is excellent and easy to use because it is highly automated. You do have to print out the on-disk documentation to make up the manual.

In sum, this level of documentation is sufficient for the more technically inclined, but is sure to leave the less informed user out in the cold. What PC Genius has done is to provide a "Personal Computer Buyer's Guide" as a substitute for complete documentation. The idea seems to be that if you are sufficiently informed to make the right buying decision, you really don't need anything more because the machine is made up of standard parts, easy to replace if they fail. Also, since the likelihood of failure is supposedly low, and since PC Genius is there if you do have problems during their 30-day money-back guarantee period, you don't really need more documentation. By the way, PC Genius has told me that they are now thinking of running classes on how to put together your own machine. This may remove the last hurdle for those of you who aren't sure that this is the way to go.

Reader Service 21

Recommendations

There is no question in my mind that buying either the Heritage 386/16 or the Darwin can be an outstandingly good deal. You shouldn't expect to find any important differences (other than documentation) in buying these machines than you would in buying a nationally recognized brand. If you find a good vendor, you can expect friendly and helpful support. At times, though, the smaller vendor may have to refer you to the supplier of a particular component. That being the case, go for a system consisting of parts made in the USA with a good reputation for customer support.

It's hard to get a better deal unless you collect the parts individually and then construct your own system. Of course, you won't have figured in your own time, which would actually raise the price considerably. Besides, the convenience of opening the shipping carton and immediately starting to use your new system cannot be overstated.

The bottom line is that if what you want is a national vendor with the options of an extended warranty and on-site service, Computer Components is an excellent buy and one that I recommend. On the other hand, I liked the value offered by the folks at PC Genius, so I heartily recommend them to those of you located in the Northeast or willing to go mail-order.

An enhanced operating environment

As a fairly typical MS-DOS engineering user, I find that my PC gets more use than the numerous other minis and mainframes that are easily available. The reason is that the PC is there, at the side of my desk, ready to give me nearly instantaneous results. It is my machine of choice for spreadsheet applications, word processing, electrical CAD, simple problem solving, and communications.

I find I don't need the timesharing or multitasking capabilities of Unix or VMS. What I do need is the ability to switch rapidly from one task to another as I go about my daily routine. In the past, this meant stopping one program on my PC and starting another with the consequent loss of context in the first program. So when DESQview for the PC came out several years ago, letting me switch from my word processor to my spreadsheet to my communications program to my database and back again, I became a committed user. This enhancement of the MS-DOS environment is more than convenient; it really adds to my productivity.

DESQview is a great program that continues to get better and better. It now has so many features that it's really hard to deal with them all in the limited space I have. As I've said, the key feature is that it lets me have several programs running at the same time and switch between them with just two keystrokes.

My primary reason for upgrading to the latest version (currently 2.01) of DESQview was to take advantage of the multimegabytes that you can have on a 386 machine. Quarterdeck's expanded memory manager (QEMM), an extra cost option, uses the 386 hardware to implement version 4.0 of the expanded memory specification (EMS). This means that you can run programs and manipulate data that reside above the 640-kilobyte limit that MS-DOS imposes. And because DV/QEMM takes advantage of the virtual 86 machine architecture of the 386, you can also run several large programs simultaneously.

As you might expect, many new features in the latest version of DESQview provide another reason for upgrading. These features include more control over program loading; better and enhanced screen support; more windows; more and better utilities for data transfer, printer support, and keystroke macros; and a new programmer's application interface. However, since this review is related to 386 systems, I'm going to limit my experiences to the DV/QEMM combination.

Installation of both DESQview and QEMM is fairly simple because of complete automation. After placing the diskette in the floppy drive and typing "install," you answer a few questions. DESQview searches a hard disk, if you have one, and automatically installs all the programs it knows about. Next you run Setup to customize the system for such things as colors, keyboard, logical drives, a pointing device, and the video monitor. Then you're done. Of course, you can change things to suit your needs. Like most users, I find that I often go back and reinstall programs or change screen colors to suit my fancy.

QEMM installation is even easier; you just type "instqemm" and answer what drive you boot from. The installation program modifies your Config.sys file to load the QEMM driver so that it is available when you start DESQview.

Once started, DESQview takes over and offers a menu of programs and other options (such as a phone dialer, a cut and paste operation, and a number of DOS utilities) for you to choose from. All of the programs I've tried run as well under DESQview as if standalone, with certain exceptions: (1) they run a bit slower, and (2) there is less memory available for a program to load into.

The combination of DOS and DESQview, plus the space needed to hold the display image, means that you won't have much more than 535 kilobytes for any one program. While this doesn't approach a serious limit for most programs, it does mean that some, like very large spreadsheets and CAD systems, cannot be run under DESQview, even when operating with QEMM.

One warning: It's not true that all 386-based systems are created equal. Of the three systems I've worked with, not one manages expanded memory in a way that QEMM can make full use of it. Thus, when using the memory status utility to examine how much expanded memory is mapped and available, each system reported about 380 to 500 kilobytes less total available expanded memory than expected. This isn't a fault of QEMM, but rather is inherent in the design of the 386 systems. As a result, you get the use of most, but not all, of the multiple megabytes that you may have on your machine.

While trying to pin down the problems with the missing memory, I used Quarterdeck's customer support line. Of all the companies I've worked with, Quarterdeck consistently has provided outstanding technical support. The support staff is very well qualified, very friendly and helpful, and exceptionally thorough.

DESQview costs $129.95, while QEMM goes for an additional $59.95. These two packages are a dynamite duo that provide nearly seamless integration of all your various and sundry—but separate—programs. I have recommended and I continue to recommend DESQview highly because it is so useful and so reasonably priced.

DESQview: **Reader Service 22**
QEMM: **Reader Service 23**

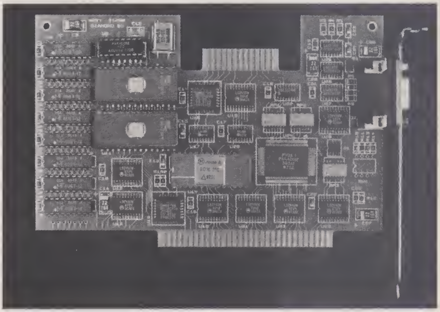
Evaluating your new PC

While evaluating 386-based machines, I was fortunate to run into Robert Brown of Landmark Software. Mr. Brown possesses a wealth of experience with PCs and has written a white paper entitled "Some Thoughts on Selecting a 386 System." If you have questions about how much memory to include with your 386, the early problems with defective 386 chips, 16- versus 20-MHz chips, and current motherboard designs using the 386 microprocessor, then I suggest you contact him. Landmark sells system boards and will help you in putting together your own, state-of-the-art system.

But Landmark Software is known much more for its Speed program. This program uses a computational algorithm that's claimed to be more representative of the work that real programs do than some of the other, well-known benchmarks. My own tests confirm this. Also, at the last Comdex, nearly everyone used Speed to demonstrate the power of their machine, whether an AT or 386. At $29.95, the package containing Speed is well worth owning.

Included with Speed is a program called Setup, which, as its name implies, will allow you to set the CMOS RAM in most ATs and AT-like 386 machines. I tried this program on all the 386 machines I tested, along with a Compaq-286, and it worked perfectly. It was also easier to use than the setup programs that came with the machines.

Reader Service 24



The Paradise VGA Plus Card from Paradise Systems incorporates the PVGA1 controller chip, which includes an 8-bit and a 16-bit data bus interface.

A striking new display adapter

As mentioned in the compatibility section, getting all the pieces to play together can be tough. For that reason, I was very pleased when Paradise offered me a prototype of the Paradise VGA Plus Card. With equivalent compatibility to that of the IBM VGA in VGA, MCGA, EGA, CGA, and MDA as well as Hercules mode, this display adapter card offered me a chance to test all of my software in combination with the 386 systems. The results were impressive, and proved very enlightening.

Before I get into the compatibility issues, let me describe the VGA Plus Card. Physically, this card is striking in two respects—it's very small, containing very few chips, using mostly surface-mount technology; and it has no switches to set, jumpers to move, or other adornments that require careful study of the user's manual. Even the feature connector of the EGA is gone; in its place is an edge connector. To use this device, you need only take it out of the box, place it in an empty slot, and boot your system. Everything else is taken care of for you automatically. I like this kind of simplicity.

In addition to the normal-modes graphic modes, you also get 800 × 600 pixels in sixteen colors and the ability to display text 132 columns wide. Drivers are included for Microsoft Windows, AutoCAD, GEM, Ventura Publisher, Cadvance, Framework, WordPerfect, and Wordstar. While I didn't get a chance to test these enhanced modes, I did use the board with DESQview, Microsoft Word, Multiplan, Procomm, Leading Edge Word Processing, and CADKEY, just to mention a few programs I use daily.

With the exception of Multiplan, everything worked flawlessly on the 20-MHz Darwin 386 system. In the case of Multiplan, I had to request the latest version (3.04) from Microsoft. It appears that there is a conflict between the software and the VGA BIOS that causes the screen to come up blank if installed in IBM-compatible mode. The reason I blame it on the BIOS is that the Paradise board was not recognized by the 386 during POST until I installed the latest Award BIOS (version 3.3) in the 386. Then everything worked as described.

To take full advantage of the features of the Paradise VGA Plus, you will need a multifrequency-type monitor. I used both NEC and Sony monitors, in the end preferring the NEC because of its on-top sizing controls.

The first thing you notice is the color saturation that this board produces. The colors are brilliant, with true yellows and browns. A second observation is that, unless you are in the higher resolution, the vertical size is smaller. While the VGA signals include a control line for the picture size, none of the monitors I used could take advantage of it. So a little adjustment of the monitor is called for. On the other hand, when I used the board with CADKEY in VGA mode, it filled the screen and provided a clean, crisp image that was noticeably better than what I was used to with my older EGA card.

Along with the board you get two complete, detailed manuals. The first manual covers installing the VGA Plus, setting DIP switches on the computer's motherboard, checking the ROM version, and using the utility program for manually switching between display modes, and, if desired, locking one mode as the default. Also included is a description of the unusual 15-pin, D-type connector that provides the video port. The second manual covers the extended 800 × 600-pixel graphics modes and the drivers that Paradise supplies with the board on a 5¼-inch floppy.

The 15-pin connector is unusual in that there is an extra row of pins in what looks like a standard 9-pin connector. You can't connect the board to your monitor using your old EGA cable because of this. Fortunately, you wouldn't want to, because the output is analog, not digital, and the pinning is different. What you need is a special cable (available at least for the NEC) that converts the 9-pin connector to its 15-pin analog equivalent. By the way, for those of you using a monitor from some other manufacturer, you will most likely not be able to use the NEC cable. I tried it with the Sony and found that it uses a different interconnection scheme from the NEC in analog mode even though both are identical in TTL mode.

The heart of the VGA Plus Card is the Paradise PVGA1 video controller chip. The chip is used in both the Plus and the Professional display adapter cards from Paradise. Because this chip includes both an 8- and a 16-bit data bus interface and uses a more efficient architecture, a 50 percent speed improvement over IBM's VGA add-on card is claimed by Paradise when using their VGA Professional Card.

The difference in price between the Plus and the Professional is $200, with the Plus retailing for $399. The Professional offers a number of additional features, including several higher resolution monochrome modes such as 1024 × 768 pixels. Higher resolution color modes are also available, including 800 × 600 with 16 colors or 640 × 480 with 256 colors out of a palette of 256,000.

Given the level of performance at a comparable price for that of an EGA display adapter, there seems to be little reason to opt for anything other than the Paradise VGA Plus. Personally, I've retired my EGA board because I prefer the vivid colors and the excellent resolution of the Paradise Plus.

Reader Service 25

Product notes

- Paradise Systems has just begun shipping their new AutoSwitch Monochrome EGA card. If you have a monochrome monitor on your PC, XT, AT, or compatible, want EGA graphics, and don't want to purchase an EGA monitor, this new display adapter might be just the thing. Running all EGA software written for color monitors, but translating colors into shades of gray, green, or amber (depending on the monitor type), this short card claims 100-percent compatibility with EGA, CGA, MDA, and Hercules graphics modes. The retail price is $249. Paradise Systems can be reached at 99 S. Hill Dr., Bisbane, CA 94005, (415) 468-7300.

- Quadram has reduced the price of its Quad 386XT to $995. The board works with a PC- or XT-compatible (but not an 8086-based system) and comes with one megabyte of 32-bit memory, Quad Virtual Monitor software for memory management, RAM drive software, print spoolers, and drivers to support the new LIM 4.0 expanded memory specification. Using a daughter card, memory can be expanded to two or eight megabytes, and the board is socketed for the 387 and 287 math coprocessors. Quadram Corp. is located at One Quad Way, Norcross, GA 30093-2919, (404) 923-6666.

- For those developers looking for a real-time operating system for Intergraph's Clipper, go to JMI Software Consultants, Inc., 904 Sheble Lane, PO Box 481, Spring House, PA 19477. This group has just added the Clipper to its large stable of CPUs, all running the same real-time operating system, called C Executive. Included are the DEC LSI-11, Motorola 6809 and 680XX series, Intel 808X and 80X86 series, Zilog Z80, Hitachi 64180, TI TMS34010, AT&T WE32100, and National 3200 series. JMI can be reached at (215) 628-0846.

- If you have a classified computer requirement, you might want to consider Iverson Technology Corp. As an IBM authorized VAR, Iverson offers PS/2 models 30, 50, and 60 (or an Intel 386 PC) with removable hard drives with capacities from 20 to 120 megabytes. Up to two drives can be combined in one system. These drives are easily removed in less than 10 seconds for either secure storage or transportation from one location to another. For further information on these systems as well as other secure computer products, contact Iverson at (703) 749-1200 or write to Iverson Technology Corp., PO Box 6070, McLean, VA 22106.

- Integrated Device Technology has introduced a number of JEDEC-compatible, one-megabit static RAM modules with a 45-nanosecond access time. Available in LCC and SOIC configurations, the megabit modules are organized as 128K × 8 or 64K × 16. IDT, 3236 Scott Blvd., PO Box 58015, Santa Clara , CA 95052-8015, (408) 727-6116.

- The V-ARC02 is a VMEbus-compatible LAN interface board that uses industry-standard ARCnet cable transceivers to connect to a wide range of other ARCnet bus interface products, including the Q-bus Q-ARC01. The products, developed by Comendec Ltd., are distributed by C&C Marketing, 708 Mandrake Drive, PO Box 280, Batavia, IL 60510, (312) 879-7003.

- Hybrid Systems is offering the HS9576 in the industry-standard ADC76 pin-out. This 16-bit ADC offer is available with full Mil-Std-883C screening and guarantees no missing codes to 14-bit accuracy from -55 °C to 125 °C. Featuring an internal reference, clock, short cycle (17-microsecond) capability, and serial or parallel output in a 32-pin ceramic DIP, this device is available from Hybrid Systems Division of Sipex Corp., 22 Linnel Circle, Billerica, MA 01821, (617) 667-8700.

THE COMPUTER SOCIETY OF THE IEEE

Membership Application

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

HOW TO JOIN

Membership dues and publication subscriptions are annualized to December 31.
Pay the full-year fee if application is mailed September 1–February 29.
Pay the half-year fee if application is mailed March 1–August 31.

1 COMPUTER SOCIETY ONLY (affiliate membership).

You are eligible if you are seriously interested in any aspect of the computer field or if you are a member of one of the affiliate societies listed below.
(Check all applicable societies.)

	Half-Year Mar 1-Aug 31	Full-Year Sept 1-Feb 29
Affiliate Societies	☐ $19.50	☐ $39.00

- ☐ American Institute of Aeronautics and Astronautics (AIAA)
- ☐ American Institute of Physics (AIP)
- ☐ American Mathematical Society (AMS)
- ☐ American Society of Mechanical Engineers
- ☐ Association for Computing Machinery (ACM)
- ☐ Australian Computer Society
- ☐ British Computer Society
- ☐ Data Processing Management Association (DPMA)
- ☐ Information Processing Society of Japan (IPSJ)
- ☐ Institute of Electronic, Information and Communication Engineers (IEICE) of Japan
- ☐ Institution of Electrical Engineers (IEE–UK)
- ☐ Instrument Society of America (ISA)
- ☐ Mathematical Association of America
- ☐ National Association of Accountants
- ☐ Operations Research Society of America (ORSA)
- ☐ Society of Aircraft Materials and Processes (SAMPE)
- ☐ Society of Automotive Engineers (SAE)
- ☐ Society for Industrial and Applied Mathematics (SIAM)
- ☐ Society for Computer Simulation (SCS)
- ☐ Society for Photo-Optical Instrumentation Engineers (SPIE)
- ☐ The Computer Society of the Republic of China (Taiwan)

2 COMPUTER SOCIETY AND IEEE.*

In addition to your Computer Society benefits, you'll receive many IEEE privileges and benefits. You are eligible if your technical interests are in computer science and engineering, the electrical and electronics fields, or related fields. Your entry membership grade will be determined by your level of participation, contributions, education and/or experience in those fields.

Residence	Half-Year Mar 1-Aug 31	Full-Year Sept 1-Feb 29
United States	☐ $41.00	☐ $82.00
Canada	☐ $38.50	☐ $77.00
Europe, Africa, Middle East	☐ $39.00	☐ $78.00
Latin America	☐ $33.50	☐ $67.00
Asia, Pacific	☐ $34.50	☐ $69.00

*ACM members who join both IEEE and the Computer Society are entitled to a deduction of $5 off the full-year rate, $2.50 off the half-year rate

3 COMPUTER SOCIETY MEMBERSHIP FOR IEEE MEMBERS.

If you are presently an IEEE member, you may join the Computer Society for a nominal amount. (Complete only shaded area of application.)

	Half-Year Mar 1-Aug 31	Full-Year Sept 1-Feb 29
IEEE Member Number ______	☐ $ 7.50	☐ $15.00

OPTIONAL PUBLICATIONS

4 Whichever membership option you chose, you are now eligible to subscribe at low member rates to these periodicals.

	(issues per year)	Half-Year Mar 1-Aug 31	Full-Year Sept 1-Feb 29
IEEE Computer Graphics & Applications (3061)	6	☐ $ 9.50	☐ $19.00
IEEE Design and Test (3111)	6	☐ $10.00	☐ $20.00
IEEE Expert (3151)	4	☐ $ 6.00	☐ $12.00
IEEE Micro (3071)	6	☐ $ 9.00	☐ $18.00
IEEE Software (3121)	6	☐ $ 8.50	☐ $17.00
Transactions on Computers (1161)	12	☐ $ 9.00	☐ $18.00
Transactions on Pattern Analysis and Machine Intelligence (1351)	6	☐ $ 7.50	☐ $15.00
Transactions on Software Engineering (1171)	12	☐ $ 9.00	☐ $18.00

TOTAL ______
(remittance in US dollars only)

☐ Check or Money Order (payable to the IEEE)

☐ Visa ☐ Master Card ☐ American Express

Credit Card Number ______ Expiration Date ______

STUDENTS! ► You receive special rates and benefits. Call or send for a student application.

I hereby make application for Computer Society membership and, if elected, will be governed by IEEE's and the Society's Constitutions, Bylaws, and Statements of Policies and Procedures.

MAILING ADDRESS Full signature ______ Date ______

First Name ______ Middle initial(s) ______ Last name ______

Street address ______ City ______ State/Country ______ Postal Code ______

Were you ever a member of the IEEE? ☐ Yes ☐ No If yes, please furnish Grade ______ Membership No ☐☐☐☐☐☐☐ Date of birth ______ ☐ Male ☐ Female

EDUCATION (highest level completed) Course ______ Degrees received ______ Date ______

Name of educational institution (Please print full name) ______

OCCUPATION Title or Position ______

Firm Name ______

Firm address ______

City ______ State/Country ______ Postal Code ______

REFERENCE (an IEEE member, if unknown, a managerial person who knows you professionally)

Name (print in full) ______ IEEE Member No (if applicable) ______

Position ______

Street Address ______

City ______ State/Country ______ Postal Code ______

PC0488

Residents of Europe mail to:

Computer Society
of the IEEE
13, Avenue de l'Aquilon
B-1200 Brussels, Belgium
Telephone: 32-2-770-21-98

Payment to the Brussels office may be made by cheques in Belgian francs, British pounds sterling, German marks, Swiss francs, or US dollars, or by American Express, Eurocard, MasterCard, or Visa credit cards.

All others mail to:

Computer Society
of the IEEE
10662 Los Vaqueros Circle
Los Alamitos, CA 90720–2578
Telephone: (714) 821-8380

Allow six to eight weeks for delivery of publications.

NEW PRODUCTS

Graphics supercomputer hits 64 Mflops

Ardent Computer (formerly Dana Computer) has announced the Titan graphics supercomputer. According to the company, the Titan uses a 64-bit parallel vector architecture to achieve a peak processing rate of 64 million floating-point operations per second.

The architecture also incorporates 52 image planes and parallel pixel processors for graphics code. This reputedly allows graphics to be processed and displayed at 200,000 full-color 3D shaded polygons per second.

Titan can be configured with up to four 64-bit vector/scalar processors paired with an equal number of general-purpose integer processors. Each processor pair shares a single instruction stream.

The design features up to 32,000 64-bit-wide vector registers, an aggregate bus bandwidth of 256M-bytes per second, and up to 128M-bytes of 16-way interleaved memory.

According to the company, Titan features 52 graphic image planes organized as 24-bit, double-buffered true color with four overlay planes. The graphics output device is a 19-inch color display with 1280 × 1024-pixel resolution and a 64-MHz, noninterlaced refresh rate.

The Unix operating system provides compatibility with AT&T Unix System V.3 and Berkeley 4.3 extensions. It includes communications and networking support for Ethernet, thin Ethernet, NFS, and other protocols, as well as C and Fortran compilers.

Ardent's Dynamic Object-Rendering Environment comes with the system.

Single- and dual-processor versions are available now, with three- and four-processor models available in the fourth quarter. Prices start at $79,000 for a single-processor Titan, up to $150,000 for a four-processor version. Field upgrades are possible, with no software changes required.

Reader Service 30

The Titan from Ardent Computer combines a 64-bit supercomputer vector architecture with interactive graphics capabilities.

Chip sets are compatible with IBM PS/2

Chips and Technologies has introduced Chips/250, a seven-chip set of system logic circuits for building microcomputer systems compatible with the IBM PS/2 Models 50 and 60. According to the company, systems designed with the chips can operate at speeds up to 16 MHz and use memory configurations of up to eight megabytes.

Chips and Technologies claims that an OEM using Chips/250 can integrate the motherboard using 68 devices, permitting smaller and less complex systems. The company also claims that matched memory implementation allows memory to run at 187.5 ns instead of 300 ns on the Micro Channel.

Chips/250 consists of 82C221 CPU and Micro Channel controller, 82C222 Advanced Memory Controller, 32C223 Advanced DMA Controller, 82C225 Data Address Buffers, 82C226 System Peripherals Controller, 82C607 Multifunction Controller, and 82C451 or 452 VGA.

The chips come in 84-pin plastic leadless chip carriers or 100-pin flat packs. In 1,000-unit quantities, Chips/250 costs $157.10 for the 10/12 MHz version and $169.50 for the 16 MHz version.

A turnkey package is also available.

Reader Service 31

Software environment assists VME system developers

Plessey Microsystems and Ready Systems have jointly developed a real-time software environment for the VMEbus market. VXCEL is an integrated real-time-Unix software environment supported across Plessey's range of commercial and military VMEbus board-level products.

VXCEL reportedly merges VRTX32 and Unix to create a platform for real-time multiprocessor applications. It also provides an environment for real-time Ada.

Reader Service 32

VAXstation provides real-time color 3D graphics

Digital Equipment Corp. and Evans & Sutherland Computer Corp. have announced the VAXstation 8000 color graphics workstation, which can reputedly manipulate complex three-dimensional objects instantaneously. The workstation will be marketed and sold by both.

The VAXstation 8000 claims a 3D vector performance of 500,000 anti-aliased vectors per second and a screen resolution equivalent to more than 8,000 × 6,000 pixels.

A complete system costs $87,710, including VMS software and a one-year, integrated hardware and software warranty with on-site hardware service. DEC plans to add the Ultrix operating system later.

Reader Service 33

Vision system targets manufacturing tasks

Applied Intelligent Systems offers the AIS-3000 vision system, an image processor used for inspection and assembly tasks in manufacturing. The company has targeted the product at OEMs and VARs.

The single-board image processor features 64 parallel processors, 256 gray levels, support for up to six RS-170 cameras, RGB color display, and image size of 384 × 488 or 512 × 488 pixels.

According to the company, image processing speed exceeds 220 million pixel operations per second. Machine vision operations occur in less than six milliseconds; Boolean operations take less than one millisecond; and erosions and dilations take less than half a second.

The AIS-3000 costs $15,000 for one unit, with volume discounts available.

Reader Service 34

Parallel system claims open architecture

The Ametek Computer Research Division has announced the Series 2010 parallel processing system. The Series 2010 was designed around a communications network based on Automatic Message Routing Device circuitry. The distributed-memory, message-passing system reportedly achieves direct communications between any two nodes regardless of their locations.

According to the company, full open architecture allows any node to interface standard VME devices. Other features include local disk storage with Unix file compatibility, vector processing capabilities for floating-point operations, graphics, multiple users, and software development tools with node-level debuggers.

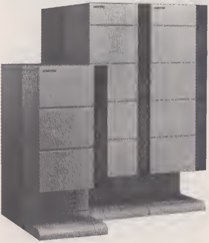
The Series 2010 can be configured with four nodes, with growth to more than a thousand nodes possible, according to the company. Pricing for a four-node system starts at $45,000. A 64-node system costs $495,000.

Each node is configured around a 25 MHz Motorola 68020 processor and comes with a Motorola 68881 Floating Point Unit. A custom memory management unit allows zero wait-state operation, reputedly insuring at least four MIPS performance from each node. Nodes can be configured separately.

The Series 2010 uses the Sun Microsystems Sun-3 workstation as the user interface. Program compilation takes place on the Sun-3, from which object code is downloaded to the 2010 for execution.

The Series 2010 was developed by Ametek and Charles Seitz and his colleagues at the California Institute of Technology.

Reader Service 35



The Ametek Series 2010 features a two-dimensional topology and the GigaLink network parallel processing architecture.

New Cray Y-MP/832 doubles performance of X-MP

Cray Research has announced the Cray Y-MP/832 computer, which the company calls an extension of the X-MP series of supercomputers. According to the company, the Y-MP/832 offers system performance two to three times that of the largest X-MP system.

The new supercomputer features eight CPUs and 32 million 64-bit words of central memory. Each CPU operates on a six-nanosecond clock cycle. It can run all applications software developed for the X-MP systems.

The Y-MP system comes standard with one Cray I/O Subsystem, with a second IOS optional. The Y-MP also comes with a solid-state storage device with 128 million words of secondary memory. Options include SSD models with 256 or 512 million words of memory.

The company's Unicos operating system, based on AT&T's Unix System V, comes on the Y-MP system. The Y-MP can also run the Cray operating system COS.

Software includes an automatic vectorizing Fortran compiler, automatic vectorizing C and Pascal compilers, a library of scientific subroutines, and a variety of system utilities.

The Y-MP system is also supported by communications hardware and software for integration with mainframe computers, workstations, and communications networks.

The Cray Y-MP/832 costs $20 million.

Reader Service 36

MicroVAX II is transportable

Tara Computer Products offers what the company calls a transportable MicroVAX II computer. The Phoenix comes in a metal case with handle. It weighs 35 to 40 pounds and measures 21 × 9.5 × 17 inches.

The Phoenix features a MicroVAX II CPU, or a PDP-11/23+ or PDP-11/73.

Memory is expandable from 256K bytes to 16M bytes. Mass storage includes an expandable 87M-byte Winchester drive, and a 400K-byte to 800K-byte RX50-compatible floppy. Tape memory includes a nine-track 1600/3200 bpi and a 60M-byte cartridge.

Operating systems include RT11, RSX, RSTS, and Micro VMS.

The Phoenix costs $19,500.

Reader Service 37

Apple tools promote host/Mac communications

Apple Computer has introduced two development tools to support cooperative processing between host systems and Macintosh computers, a hardware and software platform for development of NuBus coprocessor cards, and a new version of AppleTalk for VMS.

MacAPPC is a software implementation of IBM's Logical Unit 6.2 and Physical Unit 2.1 protocols, part of IBM's Systems Network Architecture. MacWorkstation is a software toolkit that aids in the creation of a Macintosh interface for programs operating on host computers. Version 2.0 of AppleTalk for VMS lets Macintosh and Digital Equipment's VAX systems share information across a network.

MacAPPC allows the development of Macintosh applications that can communicate as peers using the LU6.2 protocols with mainframes, minicomputers, and PCs from IBM and other vendors. MacAPPC supports the Macintosh II, Macintosh SE, and Macintosh Plus. Macintosh computers can share information with an IBM host by direct connection or over the AppleTalk network system with a Macintosh II serving as a shared SNA gateway. MacAPPC comes with HyperCard APPC for $2500 for a site license.

Mac Workstation is a development tool that lets host computer users work with Macintosh interface, filing, and printing features. The product provides programmers with access to the Macintosh toolbox. It costs $2500 under a license agreement for firms using it internally, and $5000 for commercial resale.

The Macintosh coprocessor platform is a developer building block for add-in cards for the Macintosh II, according to the company. The platform includes a Motorola 68000 processor operating at 10 MHz, 512K bytes of RAM, and a bus master interface to the NuBus. A real-time multitasking operating system supports a set of services for software executing on a card and a defined interface to the Macintosh II.

AppleTalk for VMS runs as a process under DEC's VAX VMS operating system. The software implements AppleTalk network protocols and lets VAX computers connect to Macintosh and other computers operating on an AppleTalk network. Version 2.0 provides additional protocols to implement terminal, AppleShare compatible-file, and print services. It is available under a license agreement of $5000 for a site license.

MacAPPC: **Reader Service 38**
MacWorkstation: **Reader Service 39**
Platform: **Reader Service 40**
AppleTalk 2.0: **Reader Service 41**

CAD for Mac modularized with Dreams

Innovative Data Design has announced the Dreams family of software design tools for computer-aided design on the Apple Macintosh II computers. According to IDD, the modular nature of Dreams allows users to customize the program for their specific design environment.

The basic product includes Drafting Palette, Accessory Palette, and Dimension Palette. Drafting Palette provides a set of tools for creating text, lines, or shapes. Accessory Palette provides tools for zooming, rotating objects around various axes, extending lines to their intersection, and creating fillets. It also has tools for adding or deleting handles within polygons, gluing or ungluing edges to objects, and adding or subtracting objects from each other. Dimension Palette supports both point-to-point and object-dependent dimensioning, plus automatic calculation of string dimensioning between objects.

IDD plans three optional architectural symbol libraries, including a Preliminary Design Symbol Library and two separate volumes of construction details for residential and light commercial design. Future libraries will include electrical engineering, interior design, and mechanical engineering. Users can also create their own custom symbol libraries.

According to the company, additional utilities will eventually include plotter drivers, file conversions, and 3D perspectives.

Dreams is scheduled to ship in July for $500. Add-on modules will be priced and packaged separately. Registered owners of the company's previous CAD package, MacDraft, will have the option of purchasing Dreams for $200 directly from the company during initial shipment.

Reader Service 42

Watcom offers C tools

Watcom Products has announced the Watcom C 6.0 compiler and Watcom Express C development environment.

The C 6.0 compiler's code generator permits specification of register and calling conventions to match the conventions of other C compilers and programming languages, according to the company. The compiler's programming tools include a windowed source-level debugger.

Express C reportedly provides an integrated development environment, including editor, compiler, debugger, and run-time library, all memory-resident. According to the company, programs can be compiled in memory and executed directly without separate link and load steps.

No prices given.

C 6.0: **Reader Service 43**
Express C: **Reader Service 44**

Truevision graphics card captures video for Mac

Truevision offers the NuVista video-graphics card for the Macintosh II. The card combines high-resolution video capture and display capabilities with a 32-bit graphics processor, according to the company.

The single-slot NuVista reputedly digitizes a standard video signal in real time, in full color. It also generates an analog video output signal. The card functions as an indigenous Macintosh graphics controller, and provides programmable capture and display resolutions. The 4M bytes of on-board video memory, along with a custom video cross-point chip, accommodate several memory organizations, according to the company.

NuVista provides more than 16.7 million colors. It incorporates the Texas Instruments TMS 34010 graphics processor and employs dual-ported CMOS video memory.

Other features include a programmable pixel clock, input and output lookup tables, integrated genlock (master or slave), integral and fractional zoom, panning across the entire video memory, and an eight-bit alpha channel.

End-user sales begin in June. The NuVista card will cost $5995 for the 4M-byte version and $4250 for the 2M-byte version.

Reader Service 45

Claris updates Mac software

Claris Corp. has updated Macintosh programs with its release of MacProject II, MacPaint 2.0, MacDraw II, and MacWrite 5.0.

MacProject II helps manage projects. New features include the ability to manage an unlimited number of tasks per project, the ability to link and consolidate projects, and multiple resource calendars (up to eight). Planned and actual task status are displayed in the schedule chart, Gantt charts, and project table. The software is compatible with the Macintosh 512K Enhanced, Plus, SE, and II computers. It costs $495, or $145 for an upgrade to the original MacProject.

MacPaint 2.0 features multiple windows, tear-off tool palettes, and a snapshot function for retouching. The program is compatible with the Macintosh 512K Enhanced, Plus, SE, and II computers. It costs $125, or $25 for an update.

MacDraw II runs on Macintosh 512K Enhanced, Plus, SE, and II computers. The software can read and revise all files created with the original program, according to the company. It provides on-screen color and color output for color separations. It also permits layering of up to 500 layers on a 1M-byte Macintosh. MacDraw II will be available in the second quarter of 1988 for $395, or $100 for an upgrade to registered MacDraw owners.

MacWrite 5.0 features a built-in spelling checker with a 100,000-word dictionary and the capability for users to build their own dictionaries. Other enhancements include keyboard shortcuts for cursor control and for frequently used functions, and support for large-screen monitors. The word processor is compatible with all Macintosh models from the 512K Enhanced to the Macintosh II, with System 3.2 or later, Finder 5.3 or later, and MultiFinder and AppleShare. It costs $125, or $25 for an upgrade.

MacProject II: **Reader Service 46**
MacPaint 2.0: **Reader Service 47**
MacDraw II: **Reader Service 48**
MacWrite 5.0: **Reader Service 49**

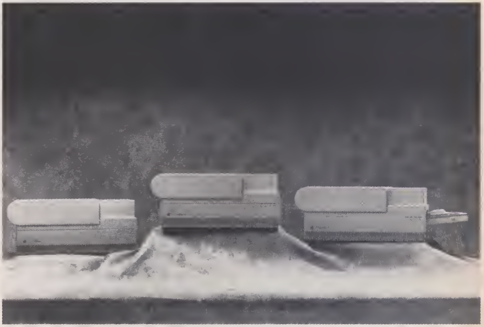
Apple LaserWriter II family replaces older printers

Apple Computer has replaced its older desktop laser printer family with the LaserWriter II family, which consists of the LaserWriter IINTX, LaserWriter IINT, and LaserWriter IISC.

The LaserWriter IINTX, based on the 16-MHz Motorola 68020, is a 300 dot-per-inch expandable laser printer with networking capabilities and a Postscript interpreter. Standard features include 1M byte of ROM, 2M bytes of RAM, 35 standard typefaces, and support for sharing the printer on AppleTalk networks. Options include memory, font expansion card, and small computer systems interface (SCSI) hard-disk drive expansion. It also provides an RS-232 serial port for connection to a host, and emulation of the Diablo 630 and Hewlett-Packard LaserJet Plus.

The LaserWriter IINT is a 300 dpi laser printer with networking capabilities and Postscript functionality. The IINT comes with 1M byte of ROM and 2M bytes of RAM. It incorporates the Motorola 68000 microprocessor, 35 standard typefaces, and built-in support for sharing the printer on AppleTalk networks. A built-in Diablo 630 emulation mode allows MS-DOS and OS/2 compatibility, and an RS-232 serial port allows connection to a host computer.

The LaserWriter IISC is an entry-level, single-user laser printer with 300 dpi full-page text and graphics. It can be upgraded to the higher performance IINT and IINTX by changing controller boards. The IISC incorporates the Motorola 68000 and has 1M byte of RAM. It comes with an SCSI interface.



The LaserWriter II family of modular laser printers from Apple Computer includes the entry-level IISC (left), IINT (middle), and Motorola 68020-based IINTX (right) with expansion and networking capabilities.

All three printers rely upon the Canon LBP-SX second-generation printing engine, reputed to have a 300,000-page lifespan. The printers weigh 45 pounds.

The LaserWriter IINTX costs $6599. The LaserWriter IINT costs $4599, and the LaserWriter IISC is $2799. Upgrades range in price from $2099 to $4299.

Reader Service 50

Compiler technology automatically generates code

NKR Research has announced the NKR Fortran compiler, which the company says incorporates automatic code-generation technology. The compiler also has built-in VAX/VMS Fortran extensions, a custom-built optimizer, and a run-time library, and reportedly meets the requirements of Mil-Std-1753.

According to the company, automatic code-generation results in faster retargeting of products to new machines and easier maintenance. Basically, the machine instruction set and architecture are described with formal grammatical rules. Changes in the machine instruction set are reflected by adjusting the description, without requiring extensive changes to internal compiler code.

NKR Fortran is available in the Unix/68K environment and can be ported to Motorola 68K-based systems. Evaluation copies are available to original equipment manufacturers.

Reader Service 51

TI adds new models to Business System series

Texas Instruments has announced new models for its Business System 600 and 800 Series of multiuser computers. According to the company, the new models—677A, 677S, 877A, 877B, and 877S—provide greater storage capacity and faster disk access at a lower price.

The new models feature a 182M-byte 5¼-inch Winchester disk using SCSI interface technology, and a 60M-byte cartridge tape backup.

The Business Systems 677A and 677S are configured with the 990/10A processor. They have 1.5M bytes of RAM, expandable to 2M bytes. The System 677A comes with a Model 931 terminal and the System 677S, with a Model 924 terminal. The 677A costs $21,495 and the 677S costs $20,995.

The Business Systems 877A, 877B, and 877S feature the 990/12 LR processor with a 1M-byte cache memory controller and a 512K memory array. All have a four-channel communications interface. The 877A and 877B come with two Model 931 terminals and the 877S comes with two Model 924 terminals. Prices are $32,995 for the 877A; $33,595 for the 877B; and $31,995 for the 877S.

TI will no longer offer the Business System models 362, 375, 673, 674, 675, 690, 691, 874, 875, and 890. The retained models have been reduced in price.

Reader Service 52

Avocet offers compilers

Avocet Systems has announced the MXPascal System, C Cross Compilers, and C Native Compilers.

The MXPascal System is available for the 68000, Z80 (NSC800 or HD64180), 8051, and 8088/86. It reportedly combines an interpreter and a compiler to enable users to write and run a program on an IBM PC or compatible, then generate code for the target system using the same source file with the same compiler system.

The MXPascal System package consists of the MXPascal compiler, MXRun run-time interpreter, a code generator for a buyer-specified target chip, and a code generator for the 8088/86 chip. MXPascal requires an editor to create standard ASCII files and the appropriate assembler: XMAC68 for 68000; AVMACZ80 for Z80, NSC800, or HD64180; AVMAC 51 for 8051 family; Microsoft MASM or Phoenix PASM for 8088/86.

MXPascal costs $395, including the code generator for one target. Extra code generators cost $149 each.

The Avocet C Cross Compilers are available for the Z80 (including NSC800 and HD64180) and the 8096. Avocet C Native Compilers are available for the 8088/86/186/286 and Z80. Compatible with Kernighan and Ritchie, the compilers offer such C features as floating point, bit fields, unsigned char and long, and an I/O library.

Avocet C Compilers generate code for execution under a host operating system or installation in ROM. A run-time library, provided in source and object form, implements standard C I/O and other functions.

Avocet C Compilers run on IBM PCs and compatibles. They come with a macro assembler and linker for $895. Registered AVMAC owners can purchase C Cross Compilers for $595 and C Native Compilers for $295.

MXPascal: **Reader Service 53**
C Compilers: **Reader Service 54**

386 Fortran compiler

Science Applications International Corp. offers SVS Fortran-386 for the Compaq Deskpro 386 and compatible computers. According to the company, the compiler allows users to develop Fortran programs up to 16M bits in size and run them in the protected mode of the 386 under the control of an existing MS-DOS (or PC-DOS) 2.x or 3.x.

SVS Fortran-386 includes a native code compiler, a run-time I/O system, a source-level debugger, and VM/Run, a 386 run-time monitor.

SVS Fortran-386, including the source-level debugger and VM/Run run-time monitor, costs $895. This includes a royalty-free license that allows users to bundle the monitor with applications.

Reader Service 55

Development system relies on Ada

Ready Systems has announced RTAda, a development system the company says was designed for real-time embedded applications. The system includes an Ada runtime system and Ada cross-development tools.

The RTAda Runtime System reportedly implements the full Ada tasking model, dynamic memory allocation, exception handling, and I/O. It connects an Ada application to the real-time multitasking facilities of Ready Systems' Ada real-time kernel, ARTX, to allow access to a set of packaged interfaces to real-time constructs such as queues, event flags, semaphores, and mailboxes. The Runtime System also provides a mechanism for integrating tasks implemented in other languages, such as C, Pascal, or Fortran, with tasks written in Ada.

The RTAda Cross Development System includes self-hosted and cross-development validated Ada compilers with support for Chapter 13 facilities. Also provided are a source-level debugger, global optimizer, language tools, and tools for downloading and generation of embedded systems.

The RTAda Development System is available for Digital Equipment Corp. VAX hosts running VMS, and targets both the 1750A and Motorola MC68000 family of microprocessors. Prices depend on the development configuration and host environment.

Reader Service 56

If you just need a drawing tool, use one of these.

If you need a design tool, use Design/2.0™

A lot of tools can help you draw. But how many can help you design and keep track of complex system models?

Meta Software introduces Design/2.0, the integrated text and graphics processor that helps you design systems, flow charts, organizational charts, computer programs, communications networks, production line process diagrams, business plans and more. Do it all faster and more efficiently than with manual tools or any other program.

Design/2.0 puts an end to time-consuming drawing and redrawing. Because it understands that objects in a diagram are related. Once you connect one object to another in a Design/2.0 diagram, it stays connected. If you move or resize an object, Design/2.0 automatically redraws all associated objects, arrows, and subordinate objects.

You can also create and edit text in your diagrams. And associate text with any object or connector. Even establish hypertext links that connect text across multiple pages.

As your model evolves, you can move detail to a subpage. In fact, you can build multi-level diagrams up to 9,999 pages. Design/2.0 automatically maintains the relationship and displays the hierarchy. So the big picture is easier to see. No matter how complex your model.

And now, trade in MacDraw or any PC drawing program for Design/2.0 by May 31, 1988 and get $25. Just send us your Design/2.0 registration card, indicating which drawing program you use. We'll mail you a check for $25. And start spending your time designing instead of drawing.

For more information about Design/2.0, call toll-free 1-800-227-4106.

Inside Massachusetts, 617-576-6920. Or return this coupon to: Meta Software Corporation, 150 CambridgePark Drive, Cambridge, Massachusetts 02140.

Name ____________________

Title ____________________

Company ____________________

Address ____________________

City ____________ State ________ Zip ________

Telephone ____________________

Meta Software

IECO 4/88

Available for IBM® PC-AT, PS/2 & close compatibles. (Requires Windows.™) $350
Available for the Macintosh™ Plus, SE, II. $250

IBM PC-AT and PS/2 are registered trademarks of International Business Machines Corporation. Windows is a trademark of Microsoft Corporation. Macintosh is a trademark and MacDraw is a registered trademark of Apple Computer, Inc. Design/2.0 is a trademark of Meta Software Corporation.

Reader Service Number 7

IC Announcements

Company, Model, Function	Comments	R.S. No.
Analog Devices ADSP-2100A DSP	A digital signal microprocessor, now available in 10 and 12.5 MHz "A" versions. Computes a 1,024-point FFT in 3 ms. AJ and AK grades are code- and pin-compatible with earlier 6 and 8 MHz (J and K) versions. Emulators and evaluation boards are available second quarter. Comes in a 100-pin PGA and 100-lead PLCC. Cost: (1,000s) $103 for 10 MHz and $133 for 12.5 MHz.	120
Analog Devices AD1139 DAC	Features voltage and current output data latches, internal reference, output amplifier, and application resistors for output ranging. Requires no user adjustments to achieve 18-bit accuracy, with ±½ LSB maximum differential and integral nonlinearity. Comes in a 32-pin triple-wide DIP. Cost: $295 (100s).	121
Honeywell HDAC51600 DAC	A 25-MHz 8-bit multiplexed DAC with five TTL/CMOS-compatible input buses. Includes voltage reference, data latches, timing and control logic, a 5:1 multiplexer, and a video DAC. Linearity to ±½ LSB. Comes in a 46-pin PGA. Cost: $98.75 (100s).	122
Integrated Device Technology FCT family additions Logic functions	Four CMOS logic functions, including three registered transceivers (IDT54/74FCT646, IDT29FCT52, IDT29FCT53) and an octal transceiver with parity (IDT54/74FCT833) with speed equivalent to the bipolar 53/74 "F" series. Come in 24-pin packages. Cost: (100s) $4.35 for IDT29FCT52 and IDT29FCT53, $4.80 for IDT54/74FCT646, $5.65 for IDT54/74FCT833.	123
Intel 82526 Serial communications IC	Implements the Controller Area Network protocol in an in-vehicle serial communications device. On-chip quasi dual-port RAM serves as a message buffer. Also includes two 8-bit ports. Samples available in 44-pin JEDEC PLCC. Production in third quarter 1988. Cost: $7.50 (10,000s).	124
Linear Technology LTC1092 ADC	A monolithic, 10-bit A/D converter that operates ratiometrically or with an absolute reference. The serial interface can present data MSB or LSB first, as required. Comes in an 8-pin plastic or ceramic DIP. Cost: $10.15 (100s) for plastic DIP.	125
Matsushita Electric 16-Mbit DRAM DRAM	A 16-megabit dynamic RAM measuring 5.4 × 17.38 mm. A single memory cell measures 3.3 square μm. Features an access time of 65 ns and memory cell capacitance of 60 fF, leading to a reduced soft error rate. Developed using a 0.5 μm rule process technology.	126
Motorola MC68184 BIC	A broadband interface controller to support MAP in real-time communications networking. Combines the functions of up to 50 SSI/MSI ICs into one 40-pin plastic DIP. Based on two-μm CMOS technology, dissipates less than 300 mW at 10-MHz clock frequency. Implements the digital portion of IEEE 802.4. Cost: $20 (100s).	127
National Semiconductor NM5100 FSRAM	A 256K-bit fast static RAM fabricated using the one-μm BiCMOS III process. Organized as 262,144 words × 1 bit. Offered in 15 ns and 18 ns speed grades. Comes in 24-pin ceramic DIP, or in a low inductance ceramic flatpack. Cost: (100s) $125 for 15 ns; $96 for 18 ns.	128
Signetics 74HC/HCT40105, 74HC/HCT7030 FIFOs	First two members of a new family of FIFO registers. 74HC/HCT40105 is a 4-bit × 16-word part; 74HC/HCT7030 is a 9-bit × 64-word part. Functionally identical otherwise. The 16-pin 74HC/HCT40105 comes in a plastic DIP or SO-16 package; the 28-pin 74HC/HCT7030 comes in a plastic DIP or plastic SOL package. Cost: (100s) $2.85 for 74HC/HCT40105, $35 for 74HC/HCT7030.	129
Standard Microsystems COM91C32 ARCnet LANT	An improved version of the COM90C32 ARCnet LAN Transceiver. Contains a 20 MHz crystal oscillator and a power-on-reset circuit. Compatible with COM90C26 LANC, HYC9068 ARCnet LAND, and HYC9058 ARCnet HIT. Available in ceramic and plastic DIP. Cost: $8.80 (1,000s) for plastic DIP.	130
Toshiba America TC120G Series Gate arrays	A series of gate arrays that use a one-μm CMOS process to achieve typical gate delays of 400 ps. Five master array sizes, ranging from to 37,392 to 129,042 raw gates. Compatible with the channel-less TC110G Series. Toshiba is accepting designs now.	131
Toshiba America MPGA Metal PGA	An IC package constructed mainly from metal. Offers low thermal resistance, low electrical impedance, and mechanical strength. Can house large chips. Comes with 120 or 144 pins, with more pin counts planned. Toshiba will begin shipping devices such as ASICs in the MPGA package in the third quarter.	132

Microsystem Announcements

Company, Model, Function	Comments	R.S. No.
American Micronics ZX386 Motherboard	An add-on motherboard that converts a Zenith model 248 into a 32-bit 80386 machine. Available in two versions, 16 or 20 MHz. Uses an Intel 80386 CPU chip, the Chips and Technologies chip set, an AMI BIOS, and an optional Intel 80387 math coprocessor. Cost: $2,595 for 16 MHz; $2,995 for 20 MHz.	135
British Telecom STEbox STEbus computer	Consists of a processor card, five-slot backplane, and 40W switched-mode power supply. Includes two software packages, a debug monitor, and a terminal emulator for development of STE systems on a PC. Comes with one of four processor boards, from an 8052 CPU up to a 16-bit 68000. Cost: £360 for 8052 up to £495 for 68000.	136
Connect A/S Micronet ARCnet card	A single-slot short card that allows IBM PS/2 Models 50, 60, and 80 users to operate under Novell's Netware network operating system. Built around the SMC 9026 controller chip. Includes a cache buffer. Comes with automatic software setup routines. Cost: $398.	137
Dataram Wide Word 2000 Memory	Rack-mountable memory units. Can be combined for a total capacity of 80G bytes. Captures up to 640M bytes in 3.2 seconds, provides a memory transfer rate of 200M bytes per second, and interfaces with up to eight I/O peripheral devices. Features one-megabit RAM technology. Cost: $76,000 for 32M bytes up to $450,000 for 640M bytes.	138
DSP Design ARC500 ARCnet card	An interface card that links IBM PS/2 Models 50, 60, and 80 to the ARCnet LAN. Features a 16-bit data path into a dual-ported memory buffer, two-kilobyte dual-ported RAM, network transceiver circuitry coupled to an isolated ground BNC connector, and programmable option select. No prices given.	139
IOtech MacSCSI 488 Interface	An Apple Macintosh interface that connects up to 14 IEEE instruments and peripherals to the Mac's SCSI port. An external unit that supports data transfers at 600K bytes per second for the Mac Plus and Mac SE, and 800K bytes per second for the Mac II. Comes with software device drivers and desk accessory software. Cost: $795.	140
Konan Microchannel Extender Extender board	A Micro Channel extender board for IBM PS/2 Models 50, 60, and 80. Extends all 116 signals in the 8-bit section and the 16-bit extension for troubleshooting and repair. Includes an adapter retainer and adapter bracket. Cost: $150.	141
Meridian Data CD Net, CD Server CD networking	CD Net, for small to medium LANs, has three half-height slots, a network board with 8088 microprocessor, integration/operation software, and power supply. It can be configured with CD ROM and hard disk drives. CD Server, for medium to large LANs, has the same features with additional computing abilities. Cost: $2,995 for CD Net; $5,995 for CD Server.	142
Motorola MVME147 VMEbus board	A VMEbus board based on the MC68030 32-bit MPU. Implements the MVME6000 VME gate array. Features an SCSI bus interface, four serial I/O ports, a Centronics printer port, an Ethernet transceiver interface, and a maximum of eight megabytes of memory. Cost: $3,747 (100s).	143
Parsytec GmbH VMTM VME processor board	A multiprocessor module for the VMEbus with four 32-bit transputer nodes. Has a software configurable link switching scheme. Serves as an accelerator module for VME applications or as a development module with OS-9 or Unix systems on VME. Cost: $8,375.	144
Tandem Computers V80 Subsystem Disk subsystem	A disk subsystem that combines the V80 Disk Storage Facility and the 3120 Disk Controller. The 3120 features the ANSI-standard intelligent peripheral interface. The V80 stores 2.7G bytes of unformatted data or 2.1G bytes of formatted data. The 3120 transfers data at 2.4M bytes per second. Cost: $62,000 for four disk drive modules; $87,000 for six; $112,000 for eight. 20,000 for the 3120.	145
Tecmar MicroRAM AD Memory board	A memory/multifunction board for the IBM PS/2. Has serial and parallel I/O capabilities. Can expand LIM EMS 4.0 compatible memory to eight megabytes for MS-DOS users. Offers up to eight megabytes of OS/2 addressable memory and compatibility with Micro Channel-based models. Cost: $1,145 with two megabytes; $200 for I/O module.	146
Tecmar MicroRAM 50/60 Memory board	Offers up to two megabytes of memory expansion for the IBM PS/2 while supporting IBM FEFE addresses. Hardware compatible with EMS 4.0. Provides an alternative to the IBM Micro Channel 80386 Memory Expansion Adapter/A. Cost: $995 with two megabytes.	147

ADVANCE ANNOUNCEMENT

The 8th
International
Conference on

Distributed Computing Systems

June 13-17, 1988 • Fairmont Hotel • San Jose, CA

Sponsored by:
The Computer Society of the IEEE and the Technical Committee on Distributad Processing
The Institute of Electrical and Electronic Engineers, Inc.

General Chair: Carl Davis
University of Alabama/Huntsville
Dept. of Computer Science
Huntsville, AL 25899
(205) 895-6088

Program Chair: Walt Kohler
University of Massachusetts
Dept. of Electrical and Computer Engineering
Amherst, MA 01002
(413) 545-0765

The 8th International Conference on Distributed Computer Systems will provide an opportunity to study the technical aspects of specifying, designing, implementing, and evaluating distributed computing systems. In such systems, there are multiple processing resources interconnected to cooperate under system-wide control with minimal reliance on centralized procedures, data, or hardware. Also, the location of computing resources may span the spectrum from physical adjacency to geographical dispersion.

The topics of scheduled to be addressed during the conference include the following aspects of distributed computing systems:

☐ Decentralized and Parallel Computer Architectures
☐ Communication Architectures and Protocols
☐ Distributed Operating Systems
☐ Applications
☐ Distributed Database Systems
☐ Languages, Tools, and Software Engineering
☐ Modeling and Performance Evaluation

Pre-conference and post-conference tutorials are being offered to provide in-depth study in the following areas:

- **Tutorial 1: Parallel Processing Networks and Systems**
  Monday, June 13, 1988
  9:00 am-5:00 pm
  Tutorial 1 will examine the design and use of tightly-coupled large-scale parallel processing systems by presenting case studies of interconnection networks, parallel algorithms, and parallel machines.
- **Tutorial 2: Tools For Developers of Distributed Software Systems**
  Monday, June 13, 1988
  9:00 am-5:00 pm
  Tutorial 2 will survey a range of tools that are, or should be, available to aid developers of distributed sofware systems.
- **Tutorial 3: Fault Tolerance Systems: Principles and Examples**
  Friday, June 17, 1988
  9:00 am-5:00 pm
  Tutorial 3 will address the factors that cause system failure (hardware faults, noise, software bugs, etc.) and will then present the wide range of techniques, both hardware and software, that have been developed to protect the system from these threats.
- **Tutorial 4: Distributed Database Management System**
  Friday, June 17, 1988
  9:00 am-5:00 pm
  Tutorial 4 will include an analysis of the Reference Model and the alternatives for the implementation of functions at each of it's layers.

Hotel Reservation Form

The Fairmont Hotel
Welcomes the 8th
International Conference
on Distributed
Computing Systems
June 13-17, 1988

Hotel: Fairmont Hotel, 170 Market Street, San Jose, CA 95113-2395, (408) 998-1900
Rates: ☐ $95.00 Single ☐ $95.00 Double

Name _______________ Affiliation _______________

City/State/Zip/Country _______________

Home Phone _______________ Work Phone _______________

A deposit equal to one night's stay or a major credit card number with expiration date is required to guarantee any reservations. All rates are subject to the applicable taxes.

Please circle the card you wish to use: AX CB DC MC V

Credit Card Number _______________ Exp. Date _______________

Arrival Date _______________ Departure Date _______________

All reservations must be received by the hotel **no later than May 13, 1988**. After this date, your reservation will be accepted on a space-available basis.

Conference-at-a-glance

TIME TABLE	Monday June 13, 1988	Tuesday June 14, 1988	Wednesday June 15, 1988	Thursday June 16, 1988	Friday June 17, 1988
7:30-8:30 am	REGISTRATION				
8:30-10:00 am		PLENARY SESSION	**4A.** Operating System Facilities for Multiprocessors **4B.** Evaluation of Protocols and Networks	**8A.** Analysis of Distributed Software Systems **8B.** Protocols for Reliable Real-Time Systems	
10:00-10:30 am		BREAK	BREAK	BREAK	
10:30 am-12:00 noon	**Tutorial 1** 9:00 am-5:00 pm Parallel Processing Networks and Systems	**1A.** Object Oriented Distributed Systems **1B.** Interconnection Architectures	**5A.** Distributed System Services **5B.** Deadlock, Commit, and Speedup Studies	**9A.** Distributed Systems Applications **9B.** Reliable Multicast Communication and Replication	**Tutorial 3** 9:00 am-5:00 pm Fault Tolerant Systems Principles and Examples
12:00 noon-1:30 pm		LUNCH	LUNCH	LUNCH	
1:30-3:00 pm	**Tutorial 2** 9:00 am-5:00 pm Tools for Developers of Distributed Software Systems	**2A.** Remote Invocation Mechanisms **2B.** Tree and Cube Connected Architectures	**6A.** Concurrent Programming Languages **6B.** Performance Studies	**10A.** Techniques For Improving Data Availability **10B.** Load Balancing	**Tutorial 4** 9:00 am-5:00 pm Distributed Database Management System
3:00-3:30 pm		BREAK	BREAK	BREAK	
3:30-5:00 pm		**3A.** Load Distribution **3B.** Analysis of Bus Structures	**7A.** Monitoring And Debugging **7B.** Distributed System Algorithms	**11A.** Analysis of Distributed Query Processing **11B.** Multiprocessor System Modeling	
5:30-6:30 pm			Distributed Processing Technical Committee Meeting		
6:00-8:00 pm		RECEPTION			

THE COMPUTER SOCIETY OF THE IEEE

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Advance Registration

Send form or facsimile to: Glenda McBride, **8th ICDCS**, Computer Society of the IEEE,
1730 Massachusetts Avenue, NW, Washington, DC 20036-1903. Make check payable to ICDCS-8.

Name ____________ Affiliation ____________

Mailing Address ____________

City/State/Zip/Country ____________

Work Phone ____________ IEEE Membership Number ____________

TUTORIALS: Please check the Tutorial(s) you wish to attend

- ☐ Tutorial 1: Parallel Processing Networks and Systems
- ☐ Tutorial 2: Tools for Developers of Distributed Software Systems
- ☐ Tutorial 3: Fault Tolerant Systems: Principles and Examples
- ☐ Tutorial 4: Distributed Database Management System

CONFERENCE REGISTRATION FEES

Advance Registration Fees—Before May 10, 1988

Conference Only: Members = $195.00 Nonmembers = $245.00 Students = $50.00

Tutorials Only: Members = $160.00 per tutorial Nonmembers = $200.00 per tutorial Students = $160.00 per tutorial

Late Registration Fees—After May 10, 1988

Conference Only: Members = $235.00 Nonmembers = $295.00 Students = $60.00

Tutorials Only: Members = $190.00 per tutorial Nonmembers = $240.00 per tutorial Students = $190.00 per tutorial

Method of Payment

☐ Personal Check ☐ Company Check

☐ VISA ☐ MasterCard ☐ American Express

Card Number ____________ Exp. Date ____________

Signature ____________

Registration fee includes the proceedings and reception. Written requests for refunds must be received in the Computer Society Office not later than May 10, 1988. Refunds are subject to a $15 processing fee

CONFERENCES

Editor: Edmund L. Gallizzi, Computer Science Dept., Eckerd College, St. Petersburg, FL 33733; (813) 864-8272; Compmail, e.gallizzi

McCarthy, Stonebraker, Hayes-Roth keynote Data Engineering Conference

K.M. Baumgartner, University of Illinois

John L. McCarthy, Michael Stonebraker, and Frederick Hayes-Roth delivered the keynote addresses that launched the three days of technical sessions at the Fourth International Data Engineering Conference in Los Angeles in February.

Benjamin Wah of the University of Illinois chaired the very successful conference, dubbed DE4, with John Carlis of the University of Minnesota serving as program chair.

McCarthy of the University of California's Lawrence Berkeley Laboratory presented a talk entitled "Knowledge Engineering or Engineering Information: Do We Need New Tools?" In it, he discussed the tools engineers use and how they could be integrated for day-to-day utility.

McCarthy noted that handbooks comprise one common set of tools. He discussed the difficulty of providing the same level of information in databases as handbooks contain. One of the key points he made was that our work in integrating these tools lies ahead of us.

Stonebraker of the Computer Science Department at UC Berkeley spoke on "Future Trends in Database Systems." He discussed the effects of the SQL language and said that standardization is currently inappropriate due to the current level of experience with the language.

Stonebraker speculated that database standards will not last because of their deficiencies and will be eliminated by aggressive vendors.

Hayes-Roth, the chief scientist at Teknowledge, delivered a talk entitled "Engineering, Organizing, and Managing Knowledge: Meeting Requirements for Practical AI Systems." In it, Hayes-Roth suggested that we are hindered by our lack of a model of knowledge and our attempts to seek solutions versus interpretations.

Hayes-Roth also noted that the size of our knowledge base and the number of users is growing exponentially, and that this trend is further complicating the situation. Computer-assisted automobile repair was cited as an example of the level of complexity of this problem.

The conference program also comprised 66 papers in 19 sessions, five panel discussions, and nine tutorials. The conference is gaining more of an international flavor, with a growing proportion of non-US papers. At DE3 in 1986, for instance, a dozen of the 72 papers submitted were written by non-North American authors; this year, 17 of the 66 papers were done by authors from Africa, Asia, and Europe.

Bharat Bhargava and John Riedl of Purdue University won the best paper award. Entitled "A Model for Adaptable Systems for Transaction Processing," the paper described an adaptable transaction processing strategy. The authors concluded their presentation by challenging attendees to attack the problem of adaptability in other aspects of distributed database system design to allow the system to react to changing performance requirements and to be more robust in the presence of failures.

Xiaolei Qian won the best student paper award for her submittal of "An Effective Method for Integrity Constraint Simplification." The paper described an algorithm for simplifying integrity constraints for more general problems than previous approaches and doing so more efficiently. (It should be noted that, at this conference, all papers, including those submitted by students, were reviewed using the same standards.)

The object-oriented database topic was popular at DE4. Two technical-paper sessions and one panel discussion focused on the subject. The panel on OOD attracted the largest audience of the five conference panels.

With Stephanie Camarata the chair and Steven Bankes, Martin Hardwick, Stonebraker, and Robert Strong serving as participants, the discussion centered on the issue of whether object-oriented databases provide the final solution to database management system problems. While such systems were deemed to be desirable, the consensus that was that they are clearly not a panacea.

Participants suggested that different types of applications require different types of database management systems due to difficulties such as integration of problems with continuous data such as terrain and weather to the object-oriented format.

Thanks to the efforts of Iris Kameny and Margorie Templeton, a vendor track was presented for the first time at this year's conference. The track featured sessions and demonstrations by representatives from CAA, Chen & Associates, DEC, IBM, Intellicorp, RTI, Servo Logic, Sun, Sybase, Tandem, Teradata, and Unisys.

A session by Alan Otis of Servo Logic on Gem Stone, one of the first available object-oriented database management systems, was particularly well received.

The presentation, "KEE Connection: A Bridge Between Relational Databases and Knowledge Systems" by Steve Sellers of Intellicorp, was also popular because of its technical depth.

At the customary "what have we learned?" panel that concluded the conference, four specifically encouraging thoughts emerged. They were the mix of database topics between traditional (concurrency control, query processing) and new interdisciplinary ones; the growing international flavor of the conference, both in attendance and in paper submissions; the value the participants found in the vendor sessions; and the feeling that the quality of the conference, while already good, is improving.

For DE5 slated February 6-10, 1989 in Los Angeles, Carlis will assume the post of general chair, and Dick Shuey will be the program chair.

Talks by five speakers set themes at three-day Compcon Spring 88

Glen Langdon, University of California, Santa Cruz

Keynoting the three days of conference activities, C. Gordon Bell, Michael Callahan, Federico Faggin, R. Andrew Heller, and John Hennessy introduced the session themes when they delivered day-opening addresses at Compcon Spring 88 March 1-3 in San Francisco.

Ken Stevens of NASA Ames was the general chair of this year's event, the 33rd International Conference of the Computer Society, and Hasan AlKhatib of Santa Clara University served as program chair.

Bell, vice president of engineering at Dana Computer, talked on "Trends in High-Performance Scientific and Engineering Computing."

Bell stated that the "toughest engineering problem around" is a user problem or a compiler problem—how we use these highly parallel machines that are being built.

The peak performance is "the number of megaflops the system is guaranteed not to exceed," and the user bears the burden of fitting the problem to the architecture, Bell said.

There are hardware limits to high performance, but they are surmountable, he insisted. Getting data in and out is becoming a problem, he said; but he advised his audience to just think of a gigabyte on a dual in-line package that plugs in a backplane.

Bell said that software is the real problem, and that we must change the way of programming. He also discussed the existence of the grayware problem and said that engineers and scientists are not properly trained to exploit the parallelism on real applications.

Bell urged the computer science community to help teach how to achieve high performance on real problems. He discussed the apathy of the computer science community relative to applications, and lamented that even much of what is currently known about programming for parallelism does not appear in print.

Bell discussed the lack of books on vector processing. There is only one book from the Argonne lab on parallel machine primitives, a paradigm for message passing and programming. He said he likes the evolutionary view: find out what's wrong with the current parallel languages and correct the problem. For parallel programming, we need to think more like an operating system that must manage the parallel parts of a program, he said.

Callahan, chief executive officer of Monolithic Memories, spoke on "Developments in Integrated Device Technologies." He predicted the gap in performance between bipolar and CMOS technologies would continue. He said people have been predicting the demise of transistor-transistor logic for many years, but it hasn't happened yet.

Faggin, president of Synaptics, presented a fresh, new look at computing in his talk, "Developments in Neural Networks." His overview of neural networks was well prepared and illustrated. In it, he pointed out the limitations of the current computer paradigms and presented neural net models as complementary paradigms to existing ones.

Talking on "The Emerging Workstation Environment," Heller, vice president and general manager of IBM's Austin, Texas, facility, said he was not shy about his opinions and he did not disappoint the audience.

Heller said the end users are the key and that applications must be end-user driven. He noted the strong impact of graphics on workstations, noting that it was an excellent application for absorbing millions of instructions per second.

He discussed the enormous opportunity in workstations and the importance of emerging standards such as Portable Operating Systems Environments, also known as POSIX. Connectivity and the ability for attachment are very important, he said.

In packaging, Heller noted next year's expiration of IBM's C4 patent and predicted the emergence of "direct chip attach" that allows for wider buses.

Looking ahead to 1991-92, he predicted the dominance of low-temperature complementary metal oxide semiconductors as being cheaper and just as fast as bipolar. RISC architectures will also dominate, Heller said, and special-purpose architectures will become more commonplace but will not become standard. Networking and distributed systems will become increasingly attractive as an alternative.

In spite of the new approaches and new things coming to dominate, Heller reiterated his notion that, in this field, "nothing ever dies." He speaks with some authority on this topic, he said, since he has not been shy about trying to kill projects.

Hennessy of Stanford University and MIPS Computer Systems gave an excellent talk on RISC processors entitled "Update on RISC Architectures."

He said that the reduced instruction set computer is mostly a design philosophy. RISC designs take the hardware strengths of high bandwidth memories and caches and marry them to the software strengths of higher level language and optimizing compiler technology. The approach creates tradeoffs that balance software needs with hardware efficiency.

The performance of a computer on a given job is directly proportional to the clock tick frequency, and inversely proportional to the total number of instructions executed per task and the average number of clock ticks per instruction executed, Hennessy indicated.

He showed that the greatest gains can be achieved by reducing the average clock ticks per instruction execution. He cited an idea of John Cocke of the IBM T.J. Watson Research Center to initiate multiple instructions per clock as a basis for some current work at Stanford. Researchers are attempting to initiate two instructions per clock tick to achieve greater gains.

RISCs, or streamlined instruction sets, have been criticized for not handling floating point operations. Hennessy does not feel that that is a problem if the architecture properly separates the integer from the floating point registers, uses a floating point coprocessor, and integrates it well with the main processor.

The tricks are to let the floating point processor look for its own stuff on the bus, rather than have it pass through the main processor, and to discontinue executions in the integer unit following a floating point exception.

Compcon has long been a favorite conference for announcing and previewing new products through technical sessions, and that tradition continued at the 1988 event.

Dana, now called Ardent, introduced the Titan personal supercomputer as a

A report on the awards presented at Compcon Spring 88 appears in the Computer Society News section of this issue of *Computer*. Plans are already under way on Compcon Spring 89; a Call for Participation advertisement appears on p. 10 of this month's issue.

personal visualization tool capable of physical modeling and image rendering.

Apollo introduced the RISC-based system called Prism.

Sony announced the NWS-1800 NEWS workstation, built around two 25MHz MC68030 microprocessors. A NEWS prototype was brought to the conference and demonstrated.

Stellar previewed its graphics supercomputer and some novel architectural features.

Silicon Graphics previewed its 4D-MP RISC-based shared memory multiprocessor design of a graphics superworkstation capable of 100,000 lighted polygons per second.

Digital Equipment previewed a Quad VAX 8700 multiprocessing system.

Cydrome made its first conference presentation on the Cydra-5, a multiprocessor wherein each processor has been optimized for a particular type of task.

Supertex previewed its S-1 minisupercomputer that executes the Cray X-MP/416 instruction set.

15th Computer Architecture Symposium to feature well-rounded agenda

Edmund L. Gallizzi, Eckerd College

A series of significant events is planned when the 15th Annual International Symposium on Computer Architecture unfolds May 30-June 3 in Honolulu. The session is being jointly sponsored by the Computer Society's Technical Committee on Computer Architecture and the ACM's SIGArch.

The symposium will feature a keynote address by David J. Kuck of the Center for Supercomputing Research and Development of the University of Illinois; a technically strong program composed of a dual track of rigorously selected research papers; two plenary panels from the perspective of senior researchers with widely varying approaches probing future architectures and future technologies to support architectures; presentation of the annual CS/ACM Eckert-Mauchly Award for outstanding contributions to the computer-architecture field; and a number of full- and half-day tutorials.

H.J. Siegel of the Supercomputer Research Center is the conference general chair. Doug De Groot of Texas Instruments and Yale N. Patt of the University of California at Berkeley are the program co-chairs.

In his address, Kuck will discuss this decade's successes in parallel architectures that are now providing real performance enhancements. He will also describe how these successes are harbingers of still more parallel architectural developments in the future. Previously, these performance increases had only come from the supporting technology.

The panel on "Future Technologies," chaired by C. Lee Giles of the Air Force Office of Scientific Research, will concentrate on exciting technologies that are expected to be the supporting technologies of future computers. The panel topics will include "Superconductivity," presented by Bidock Bedard of the US Department of Defense; "Optical Computing Architectures," presented by Larry Jackel, AT&T Bell Laboratories and featuring a look at the potential of high-performance computer systems supported by these technologies; "3-D Memory," presented by Alan Huang, AT&T Bell Laboratories, including a discussion of memories that have the potential of providing 10^{12} bits in a one-cubic-centimeter chip; and "Electronic Neural Networks," which will focus on an approach to parallel computer architectures.

Identifying the most likely trajectory for future parallel computer architectures will be the primary objective of the panel on "Future Directions in Parallel Computer Architecture," according to its chair, Andre van Tilborg of the Office of Naval Research.

The panelists are well known for their work in advocating particular styles of parallel computing. They are Jack Dennis (MIT), H.T. Kung (Carnegie-Mellon University), Burton Smith (Tera Computer), and Harold Stone (IBM T.J. Watson Research Center and the Courant Institute).

Fundamental issues to be addressed during the session include shared vs. partitioned-memory architectures, fine vs. coarse-grain parallelism, and special vs. general-purpose parallelism. In addition, connectionist models of parallel computing will be assessed in relation to their practicality in the foreseeable future.

Although consensus is perhaps not a likely outcome of this session, solid arguments in support of various models of parallel computing are assured.

The advertisement that appears on pp. 94-98 of this issue of *Computer* contains registration information on the symposium.

18th Multiple-Valued Logic Symposium slated for Spain in May

Michel Israël
Chair, Multiple-Valued
Logic Technical Committee

Four invited papers will be presented when the Multiple-Valued Logic Technical Committee of the Computer Society conducts the 18th International Symposium on Multiple-Valued Logic in Palma de Mallorca, Spain, May 24-26.

The invited papers are entitled "Tutorial on 20 Years of MVL Switching Theory" by S.L.Hurst; "The Application of Temporal Logic to Hardware Design" by Mashoto Fujita; "Degree of Untruth" by David Miller; and "Artificial Neural-Nets and Models of MVL" by Yoh Han Pao.

In addition, 90 submitted papers will be presented at the symposium, including 36 written by authors from North America, 38 from Europe, and 16 from Asia. All carefully refereed for technical quality and substance, they cover devices, VLSI, circuit design techniques, probabilistic and variable-valued systems, fuzzy logic, and automated reasoning.

The sponsoring MVL-TC is an international organization of 500 scientists and engineers responsible for promoting information exchange as well as fostering research in new and emerging areas within the multiple-valued logic field.

Activities include symposia, workshops, newsletters, special issues, news announcements, and so forth.

Because the membership of the TC is international in nature, intrinsic problems exist. For instance, conference sites must be chosen carefully to serve the needs of researchers throughout the world.

Nonetheless, this situation has presented special opportunities and the TC has encouraged international effort by groups of researchers addressing the same problems. Examples include ongoing work between representatives of Canada and the US, Canada and China, Holland and the US, and Japan and Germany.

Complete information about the symposium, the MVL-TC, and the committee's activities may be obtained by consulting the advertisement that appears on p. 134 of this issue of *Computer*.

CALENDAR

April 1988

CompEuro 88, Apr. 11-15, Brussels. Contact Jacques Tiberghien, Vrije Universiteit Brussel, Pleinlaan 2, B 1050 Brussels, Belgium; phone 32 (02) 641-2905.

ICSE 88, 10th International Conference on Software Engineering (ACM), Apr. 11-15, Raffles City, Singapore. Contact Tan Chin Nam or Lim Swee Say, National Computer Board, 71 Science Park Dr., NCB Building, Singapore 0511; phone (65) 772-0405.

29th Electronics Representatives Assoc. Conference, Apr. 13-18, Washington, DC. Contact ERA, 20 E. Huron St., Chicago, IL 60611; (312) 649-1333.

VHDL Users' Workshop, Apr. 17-20, Charlottesville, Va. Contact Ron Waxman, University of Virginia, Thornton Hall, Charlottesville, VA 22901.

Information Systems Conference (ASM), Apr. 17-20, San Diego, Calif. Contact Assoc. for Systems Management, 24587 Bagley Rd., Cleveland, OH 44138; (216) 243-6900.

1988 IEEE Symposium on Security and Privacy, Apr. 18-20, Oakland, Calif. Contact Dave Bailey, US Dept. of Energy, Production Operations Division/CIM Office, PO 5400, Albuquerque, NM 87115; (505) 846-4600.

Eastern Simulation Conference (SCS), Apr. 18-21, Orlando, Fla. Contact Society for Computer Simulation, PO 17900, San Diego, CA 92117; (619) 277-3888.

Workshop on Factory Communications, Apr. 19-20, Gaithersburg, Md. Contact Alfred C. Weaver, Dept. of Computer Science, University of Virginia, Charlottesville, VA 22903; (804) 979-7529.

11th IEEE Workshop on Design for Testability, Apr. 19-22, Vail, Colo. Contact T.W. Williams, IBM Corp., PO 1900, Dept. 67A/021, Boulder, CO 80301-9191; (303) 924-7692.

International Conference on Electronic Publishing, Document Manipulation, and Typography (INRIA), Apr. 20-22, Nice, France. Contact INRIA, Service des Relations Exterieures, Bureau des Colloques, Domaine de Voluceau, Rocquencourt, BP 105, F-78153 Le Chesnay Cedex, France.

Second International Conference on Expert Database Systems, Apr. 25-28, Tysons Corner, Va. Contact Edgar H. Sibley, George Mason University, ICSE Dept., 4400 University Dr., Fairfax, VA 22030.

Working Conference on Parallel Processing (IFIP), Apr. 25-27, Pisa, Italy. Contact Michel Cosnard, Laoratoire TIM3, Institut National Polytechnique de Grenoble, 46 Avenue Felix Viallet, 38031 Grenoble Cedex, France; phone (33) 7643-3726.

IEEE International Conference on Robotics and Automation, Apr. 25-29, Philadelphia. Contact Harry Hayman, 738 Whitaker Ter., Silver Spring, MD 20901; (301) 434-1990.

Strategic Teletechnologies Conference, Apr. 26-27, St. Louis. Contact Anthony B. Lake, Telematics Group, Peat Marwick Main & Co., 1010 Market St., St. Louis. MO 63101; (314) 444-1507.

Second Expert Systems Conference (ESD, SMI), Apr. 26-28,, Dearborn, Mich. Contact Engineering Society of Detroit, 100 Farnsworth, Detroit, MI 48202; (800) 457-9504.

Sococo 88, Symposium on Software for Computer Control (IFAC, IFIP), Apr. 26-28, Johannesburg, South Africa. Contact IFIP, 3 rue de Marche, CH-1204, Geneva, Switzerland.

1988 British Electronics Week (British Information Services), Apr. 26-28, London. Contact Evan Steadman Services Ltd., The Hub, Emson Close, Saffron Walden, Essex CB10 1HL England; phone (44) 799-26699.

Second Parallel Processing Symposium (California State University at Fullerton), Apr. 28-30, Fullerton, Calif. Contact Larry Canter, 1619 N. Hale, Fullerton, CA 92631; (714) 738-3414.

May 1988

Machine Vision Hands-On Workshop, May 2-3, Atlanta. Contact Constantin Soulakos, Georgia Tech Research Institute, O'Keefe Bldg, Rm. 40, Atlanta, GA 30332; (404) 894-3412.

AutoCAD Expo 88, May 2-5, Chicago. Contact Autodesk, Inc., 2320 Marinship Way, Sausalito, CA 94965; (415) 332-2344.

Second Military Computing Conference, May 3-5, Anaheim, Calif. Contact Military Computing Conference, PO 428, Los Altos, CA 94023; (415) 494-2800.

Artificial Intelligence and Advanced Computer Technology Conference and Exhibition (SCS), May 4-6, Long Beach, Calif. Contact Tower Conference Management Co., 800 Roosevelt Rd., Glen Ellyn, IL 60137-5835; (312) 469-3373.

Second European Computerized Corporate Publishing Exhibition and Congress, May 4-7, Frankfurt, West Germany. Contact Rita Kotch, Kotch & Poliak, Inc., 708 Third Ave., New York, NY 10017; (212) 557-6950.

19th Pittsburgh Conference on Modeling and Simulation (IEEE, ISA, SCS), May 5-6, Pittsburgh. Contact William G. Vogt or Marlin H. Mickle, Modeling and Simulation Conference, 348 Benedum Engineering Hall, University of Pittsburgh, Pittsburgh, PA 15261.

Workshop on Parallel and Distributed Debugging (ACM), May 5-6, Madison, Wis. Contact Bart Miller, Computer Sciences Dept., University of Wisconsin, 1210 W. Dayton St., Madison, WI 53706; (608) 263-3378.

EMC Expo 88, May 10-12, Washington, DC. Contact EMC Expo 88, PO D, Gainesville, VA 22065; (703) 347-0030.

Fourth International Software Process Workshop, May 11-13, Devon, England. Contact Leon Osterweil, Computer Science Dept., University of Colorado, Campus Box 430, Boulder, CO 80309; (303) 492-8787.

Fifth Workshop on Real-Time Operating Systems, May 12-13, Washington, DC. Contact John A. Stankovic, Dept. of Computer and Information Science, University of Massachusetts, Amherst, MA 01003; (413) 545-0720.

Software Maintenance Assoc. Conference, May 15-18, Chicago. Contact Suzanne Grenoble, SMA, Box 391432, Mountain View, CA 94039; (408) 730-1132.

CHI 88, Conference on Human Factors in Computing Systems (ACM), May 15-19, Washington, DC. Contact Gail A. Chumura, 5214 Monroe Dr., Springfield, VA 22151; (703) 750-9401.

Third International Conference on Supercomputing, May 15-20, Boston. Contact Lana P. Kartashev, International Supercomputing Institute, Suite B-309, 3000 34th St. South, St. Petersburg, FL 33711; (813) 866-2694.

Computers and Communications in the Healthcare Industry, May 16-17, Chicago. Contact Carol Every, Frost & Sullivan, Inc., 106 Fulton St., New York, NY 10038; (212) 233-1080.

Second Symposium on Space C^3 (AFCEA), May 16-19, Annapolis, Md. Contact Randy Crawford/DQ, IIT Research Institute, 185 Admiral Cochrane Dr., Annapolis, MD 21401; (301) 224-2295.

NATO Advanced Research Workshop on Highly Redundant Sensing in Robotic Systems, May 16-21, Tuscany, Italy. Contact Julius Tou, Center for Information Research, University of Florida, Gainesville, FL 32611; (904) 335-8018.

CIPS Congress 88, May 18-20, Toronto, Canada. Contact Canadian Information Processing Society, 243 College St., 5th Floor, Toronto, Ontario, M5T 2Y1; phone (416) 593-4040.

1988 International Industrial Engineering Conference, May 22-25 Orlando, Fla. Contact IIE Registrar, 25 Technology Park/Atlanta, Norcross, GA 30092; (404) 449-0460.

41st SPSE Conference, May 22-27, Washington, DC. Contact Society for Imaging Science and Technology, 7003 Kilworth Lane, Springfield, VA 22151.

International Conference on Rural Telecommunications (IEE), May 23-25, London. Contact Conference Services Dept., Institution of Electrical Engineers, Savoy Pl., London WC2R OBL, UK; phone (44) 01-240-1871.

Third International IEEE Conference on Ada Applications and Environments, May 23-26, Manchester, N.H. Contact Derek S. Morris, Dept. of Electrical Engineering and Computer Science, Stevens Institute of Technology, Hoboken, NJ 07730; (210) 420-5606.

SID 88, May 23-27, Anaheim, Calif. Contact James N. Price, Naval Ocean Systems Center, Attn.: Code 713, San Diego, CA 92152; (619) 225-2665.

18th International Symposium on Multiple-Valued Logic, May 24-26, Madrid, Spain. Contact Enric Trillas, Investigaciones Cientificas, Serrano 117, 28006-Madrid, Spain; phone 34 (91) 621-6264.

CG International 88 (CGS, BCS), May 24-27, Geneva. Contact N. Magnenat-Thalmann, MIRALab HEC, 5255 Decelles, Montreal, Canada H3T 1V6.

SIGMetrics 88, Conference on Measurement and Modeling of Computer Systems (ACM), May 24-27, Santa Fe, N.M. Contact Rebecca Koskela, SIGMetrics 88, Mail Stop B265, LANL, Los Alamos, NM 87545; (505) 667-8887, or Connie U. Smith, L&S Computer Technology, 1114 Buckman Rd., Santa Fe, NM 87501; (505) 988-3811.

International Conference on Systolic Arrays, May 25-27, San Diego. Contact Keith Bromley, Code 741-T, Naval Ocean Systems Center, 271 Catalina St., San Diego, CA 92152; (619) 225-7008.

International Workshop on Artificial Intelligence for Industrial Applications (IEEE, SICE), May 25-27, Hitachi, Japan. Contact Takao Sasayama, Technology Planning Group, Hitachi America Ltd., 50 Prospect Ave., Tarrytown, NY 10591-4698; (914) 332-5800, or Kotaro Hirawawa, 10th Dept., Hitachi Research Laboratory, Hitachi, Ltd., 4026 Kuji-cho Hitachi-shi, 319-12, Japan; (02) 94-52-5111.

15th IFAC Workshop on Real-Time Programming, May 25-27, Valencia, Spain. Contact WRTP 88, Grupo de Informatica Industrial—DSIC/DISCA, Universidad Politecnica de Valencia, PO 22012, E-46071 Valencia, Spain; phone (34) 6-360-4041.

Eurocrypt 88, Workshop on the Theory and Application of Cryptologic Techniques (IACR), May 25-27, Davos, Switzerland. Contact Paul Schobi, Ltd., Althardstr. 70, CH-8105 Regensdorf, Switzerland.

Seventh Transportable Applications Executive Users Conference, May 25-27, College Park, Md. Contact TAE Office, Code 635, NASA/Goddard Space Flight Center, Greenbelt, MD 20771; (301) 286-6034.

Document Management Systems Conference, May 25-27, Stockholm, Sweden. Contact Information Management Congress, 345 Woodcliff Dr., Fairport, NY 14450; (716) 383-8330.

Hardware and Software Conference for Real-Time Process Control (IFIP, IFAC), May 30-June 1, Warsaw, Poland. Contact Krystyna Ornatkiewicz, Congress Bureau ORBIS, PO 146, PL-00-950, Warszawa, Poland; phone (48) 22-261-668.

15th International Symposium on Computer Architecture (ACM), May 30-June 3, Honolulu. Contact H.J. Siegel, Supercomputing Research Center, 4380 Forbes Blvd., Lanham, MD 20706; (301) 731-3700.

Avignon 88, 8th International Workshop on Expert Systems and Applications (ECCAI), May 30-June 3, Avignon, France. Contact Jean-Claude Rault, EC2, 269-287, rue de la Garenne, 92000 Nanterre, France; phone (33) 1-47-807-000.

June 1988

SIGMOD 88, Conference on Management of Data (ACM), June 1-3, Chicago. Contact Dina Bitton, Dept. of Electronic Engineering and Computer Science, University of Illinois, PO 4348, Chicago, IL 60680, (312) 996-5492, or Peter Scheuermann, Dept. of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60201; (312) 491-7141.

First International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, June 2-3, Tullahoma, Tenn. Contact Richard Roberds, University of Tennessee Space Institute, Tullahoma, TN 37388; (615) 455-0631.

Conference on Computer Vision and Pattern Recognition, June 5-9, Ann Arbor, Mich. Contact Ramesh Jain, Dept. of Electrical Engineering and Computer Science, 3215 EECS Bldg., University of Michigan, Ann Arbor, MI 48109-2122; (313) 763-0387.

Enterprise Networking Event 88 International (SME), June 5-9, Baltimore, Md. Contact Society of Manufacturing Engineers, 1 SME Dr., Dearborn, MI 48121.

Third Israel Conference on Computer Systems and Software Engineering (Israel Section of the Computer Society and Information Processing Assoc. of Israel), June 6-7, Tel Aviv. Contact Jonah Z. Lavi, PO 50432, Tel Aviv 61500, Israel; phone (972) 3-664825.

Vision Interface 88, June 6-10, Edmonton, Canada. Contact Wayne A. Davis, Dept. of Computing Science, University of Alberta, Edmonton, Alberta T6G 2H1 Canada; phone (403) 432-3976.

Second International Conference on Vector and Parallel Computing (SIAM), June 6-10, Tromso, Norway. Contact Berit Hilt, Bergen Scientific Center, Allegaten 36, 5000 Bergen, Norway.

ISCAS 88, IEEE International Symposium on Circuits and Systems, June 7-9, Espoo, Finland. Contact Pekka Heinonen, Tampere University of Technology, Computer Systems Laboratory, PO 527, SF-33101 Tampere, Finland; or Olli Simula, Helsinki University of Technology, Dept. of Technical Physics, SF-02150 Espoo 15, Finland.

ECCE 88, European Conference on Communication and Enterprise (AFCET), June 7-10, Paris. Contact Conference Secretariat, ECCE 88, 156, boulevard Pereire, 75017 Paris, France; phone (33) 1-47-662-419.

Eighth International Symposium on Protocol Specification, Testing, and Verification (IFIP), June 7-10, Atlantic City, N.J. Contact Sudhir Aggarwal, Bell Communications Research, 435 South St., Morristown, NJ 07960; (201) 829-4477; or Krishan Sabnani, AT&T Bell Laboratories, 600 Mountain Ave., Murray Hill, NJ 07974; (201) 582-5740.

26th Assoc. for Computational Linguistics Meeting, June 7-10, Buffalo, N.Y. Contact William J. Rapaport, Dept. of Computer Science, State University of New York, Buffalo, NY 14260; (716) 636-3193.

Symposium on Engineering of Computer-Based Medical Systems, June 8-10, Minneapolis, Minn. Contact John M. Long, Dept. of Surgery, University of Minnesota, 2829 University Ave. SE, Suite 408, Minneapolis, MN 55414; (612) 627-4850.

SECOND ANNUAL PARALLEL PROCESSING SYMPOSIUM

CO-SPONSORED BY THE ORANGE COUNTY COMPUTER SOCIETY CHAPTER
OF THE IEEE AND CALIFORNIA STATE UNIVERSITY FULLERTON

UNIVERSITY CENTER
CALIF STATE UNIVERSITY,
FULLERTON, CA 92634

APRIL 28-30, 1988
THURS-FRIDAY-SATURDAY
8:30AM—5:30PM

THURSDAY
HARDWARE

Dr. Hecht-Nielsen*
Hecht-Nielsen Neurocomputers, Corp.
Neurocomputers

Pradid Srimani
A Heuristic Approach to Distributed Deadlock Detection and its Performance Evaluation

Dr. Jackson
A Massively Parallel Computer Designed Around the DSP56000

Dr. Canter, CSUF
A Survey of Parallel Processing Hardware

Dr. Tuazon, CSUF
Hypercube

Dr. Manzouil, S.I.U.
Fast Adders

Dr. P. Kumar, USC
Optical Arrays for Parallel Processing

Jin-Chin Wang
Michel Dubois
USC
A Performance Comparison of Cache Coherence Protocol

Paul Fargotstein, HAC
Signal Processing

FRIDAY
SYSTEMS SOFTWARE

Dr. Seitz*, Cal Tech
System Software for Hypercube

Z. Chen, LSU
Coupling Numerical Pattern Reasoning and Symbolic Reasoning in Moderately Parallel Systems

Richard Stein, RMS-SW
Essential Software for Distributed Realtime Multiprocessor Systems

Dr. Clymer, CSUF
Hypercube Multicomputer Simulation

Prof. Barbour
University of Illinois
Net Partitioning

Binay Slugla
Bell Labs, New Jersey
Techniques for Routing

H. Liu, JPL, USC
Load Balancing

Ed Andert
Automatic Task Allocation

Dr. Danhof, SIU
Petri Network

SATURDAY
APPLICATIONS

Dr. Robert Bunker, JPL
The MAX System

Chin-Hw Lee
Simulation of Systolic Array in Graphics

Kenneth Goo
Pattern Recognition, Text Search Parallel Processor

Dr. Michail Kayat
Two Applications Utilizing Parallel Processing Flight Simulation and Visualization

Dr. Saeb, CSUF
Finite Element Analysis Employing Distributed Array Processors

Dr. Orin Marvel
Mapping Applications to the System

Dr. Yeh, CSULB
Systolic Arrays for Fourier Transforms

Intel Corp.
Demonstation

Alliant Computer
Demonstration

**Three Hour Presentations*

Early Registration:
Fee:Computer Society/ IEEE/ACMMember:
$125 all or $50/day
Non-Members: $175 all or $65/day
Full Time Students: $25 all or $15/day
Proceedings: $25
Make Check Payable to:
IEEE O.C. Computer Society
At the door: Add 10%

Symposium Registration Form
Name_______________

Address_______________

____

Company_______________
Telephone_______________

Please Send Check to:
Dennis Osgard,Tres.
17420 Olive Tree Dr.
Yorba Linda, Ca. 92686 714 762 2503

Symposium Chairman:
Dr. Larry Canter 714 738 3414

THE COMPUTER SOCIETY ®

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. ®

Eighth International Conference on the Analysis and Optimization of Systems (INRIA), June 8-10, Antibes, France. Contact INRIA, Service des Relations Exterieures, Bureau des Colloques, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France; phone 33 (1) 39-635-600.

DAC 88, 25th ACM/IEEE Design Automation Conference, June 12-15, Anaheim, Calif. Contact Pat Pistilli, MP Associates, Inc. 7490 Clubhouse Rd., Suite 102, Boulder, CO 90301; (303) 530-4333.

ICC 88, International Conference on Communications (IEEE), June 12-15, Philadelphia. Contact Gary Ridge, Bell of Pennsylvania, 15th Floor, 1 Pkwy., Philadelphia, PA 19102; (215) 466-8709; or ICC-88, c/o IEEE Office, Moore School of Electrical Engineering, Rm. 209, University of Pennsylvania, Philadelphia, PA 19104; (215) 898-8106.

Third International Workshop of the Bellman Continuum (INRIA), June 13-14, Sophia-Antipolis, France. Contact A. Blaquiere, Universite de Paris VII, Laboratoire d'Automatique Theorique, Tour 14-24, 2 Place Jussieu, 75251 Paris Cedex 05, France.

Fifth International Conference on Testing Computer Software (DPMA), June 13-16, Washington, DC. Contact US Professional Development Institute, 1734 Elton Rd., Suite 221, Silver Spring, MD 20903; (301) 445-4400.

Eighth International Conference on Distributed Computing Systems, June 13-17, San Francisco. Contact Carl Davis, Dept. of Computer Science, University of Alabama, Huntsville, AL 35899; (205) 895-6088.

17th Mumps Users' Group Meeting, June 13-17, New Orleans. Contact Mumps Users' Group, 4321 Hartwick Rd., Suite 510, College Park, MD 20740; (301) 779-6555.

Third Ada Software Engineering Education and Training Symposium, June 13-17, Denver, Colo. Contact Sue S. O'Neill, Ada Information Clearinghouse, IITRI, 4550 Forbes Blvd., Suite 300, Lanham, MD 20706; (703) 685-1477.

Third Conference on Structure in Complexity Theory, June 14-17, Washington, DC. Contact Alan L. Selman, College of Computer Science, 161 CN, 360 Huntington Ave., Boston, MA 02115; (617) 437-8688.

Prolomat 88, Seventh International Conference on Software for Discrete Manufacturing (IFAC), June 14-17, Dresden, West Germany. Contact IFIP, 3 rue de Marche, CH-1204, Geneva, Switzerland.

NECC 88, Ninth National Educational Computing Conference (ACM), June 15-17, Dallas. Contact NECC 88, International Council for Computers in Education, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905; (503) 686-4414; or James L. Poirot, Computer Science Dept., North Texas State University, PO 13886, Denton, TX 76203-3886; (817) 565-2818.

IEEE International Symposium on Information Theory, June 19-24, Kobe, Japan. Contact Shu Lin, Dept. of Electrical Engineering, University of Hawaii at Manoa, Holmes Hall 483, 2540 Dole St., Honolulu, HI 96822; or Suguru Arimoto, Faculty of Engineering Science, Osaka University, Toyonaka, Osaka 560, Japan.

1988 Summer Unix Technical Conference and Exhibition, June 20-24, San Francisco. Contact Usenix Assoc. Conference Office, PO 385, Sunset Beach, CA 90742; (213) 592-1381.

International Conference on Private Switching Systems and Networks (IEE), June 21-23, London. Contact Conference Services Dept., Institution of Electrical Engineers, Savoy Pl., London WC2R OBL, UK.

18th International Symposium on Fault-Tolerant Computing, June 27-30, Tokyo. Contact Yasuo Komamiya, 2-4-8 Kikuna, Kohoku-Ku, Yokohama 222, Japan; phone (81) 044-911-8181.

Fifth International Conference on Dielectric Materials, Measurements, and Applications, June 27-30, Canterbury, England. Contact Conference Services Dept., Institution of Electrical Engineers, Savoy Pl., London WC2R OBL, UK.

Third International Conference on Data and Knowledge Bases, June 28-30, Jerusalem. Contact Databases Conference Secretariat, PO 29313, Tel Aviv 65121, Israel; phone (972) 03-654-541.

Aegean Workshop on Computing, June 28-July 1, Corfu Island, Greece. Contact John Reif, Computer Science Dept., Duke University, Durham, NC 27705; (919) 684-3048.

July 1988

1988 International Conference on Supercomputing (ACM, INRIA), July 4-8, Saint Malo, France. Contact C.D. Polychronopoulos, Center for Supercomputing, University of Illinois. 104 S. Wright St., Urbana, IL 61801 (in North and South America); W. Jalby, INRIA, BP 105, 78153 Le Chesnay Cedex, France (in Europe, Middle East, and Africa); or H. Terada, Dept. of Electronic Engineering, Osaka University, Yamadaoka, 2-1 Suita, Osaka, Japan 565 (in Japan and the Far East).

Conference on the Role of Artificial Intelligence in Databases and Information Systems (IFIP), July 4-8, Canton, China. Contact Robert Meersman, Infolab, K.U. Brabant, Postbus 90153, NL-5000 Le Tilburg, The Netherlands; phone 31 (13) 662-430.

Eighth International Conference in Computer Science, July 4-8, Santiago, Chile. Contact Alberto O. Mendelzon, CSRI, University of Toronto, 10 King's College Rd., Toronto, Canada M5S 1A4; phone (416) 978-2952.

Second Conference on Software Engineering (IEE, BCS), July 11-15, Liverpool, England. Contact Conference Services, IEE, Savoy Place, London WC2R OBL, England, UK.

CASE 88, Second International Workshop on Computer-Aided Software Engineering (ACM), July 12-15, Cambridge, Mass. Contact Pamela Meyer, CASE 88, c/o Index Technology Corp. One Main St., Cambridge, MA 02142; (617) 494-8200, ext. 1988.

12th IMACS World Congress on Scientific Computation, July 18-22, Paris. Contact Jean-Marc David, Laboratoires de Marcoussis, Computer Science Division, Route de Nozay, 91460 Marcoussis, France.

Second Workshop on Software Testing, Verification, and Analysis, July 19-21, Banff, Canada. Contact Lee White, Dept. of Computer Science, University of Alberta, Edmonton, Alberta, Canada T6G 2H1; (403) 432-4589.

International Workshop on VLSI for Artificial Intelligence, July 20-22, Oxford, England. Contact Jose G. Delgado-Frias or Will R. Moore, Dept. of Engineering Science, University of Oxford, Parks Rd., Oxford, OX1 3PJ, England, UK.; phone (44) 0865-273-188.

ICNN 88, IEEE 1988 International Conference on Neural Networks, July 24-27, San Diego. Contact Nomi Feldman, IEEE ICNN 88, 3770 Tansy St., San Diego, CA 92121; (619) 453-6222.

ECCE 88, European Conference on Computers and Education, July 24-29, Lausanne, Switzerland. Contact IFIP, 3 rue de Marche, CH-1204, Geneva, Switzerland.

Summer Computer Simulation Conference (SCS), July 25-27, Seattle. Contact Society for Computer Simulation, PO 17900, San Diego, CA 92117; (619) 277-3888.

Navy Micro/OA 88 Conference, July 25-28, San Diego. Contact Code 31.4, NARDAC San Diego, NAS North Island, Bldg. 1482, San Diego, CA 92135-5110; (619) 437-7013.

International Computers in Engineering Conference and Exhibition (ASME), July 31-Aug. 3, San Francisco. Contact ASME Meetings Dept., 345 E. 47th St., New York, NY 10017; (212) 705-7788.

August 1988

SIGGraph 88, 15th Conference and Exhibition on Computer Graphics and Interactive Techniques, Aug. 1-5, Atlanta. Contact Adele Newton, University of Waterloo, Dept. of Computer Science, Waterloo, Ontario, Canada N2L 3G1; (519) 888-4534.

1988 International Computers in Engineering Conference (ASME), Aug. 7-11, San Francisco. Contact Edward M. Patton, US Army Ballistic Research Lab, Aberdeen Proving Grounds, MD 21005.

Structured Development Forum X (Lawrence Livermore National Laboratory), Aug. 8-11, San Francisco. Contact Ted Michels, LLNL, PO 808, L-308, Livermore, CA 94550; (405) 423-4660.

Third International Conference on Applications of Artificial Intelligence in Engineering, Aug. 8-12, Stanford, Calif. Contact R. Adey, Computational Mechanics Institute, 25 Bridge St., Billerica, MA 01821; (617) 667-7582.

Seventh Conference of the Molecular Graphics Society, Aug. 10-12, San Francisco. Contact Molecular Graphics Conference Office, 16951 Pacific Coast Hwy., PO 385, Sunset Beach, CA 90742; (213) 592-1382.

31st Midwest Symposium on Circuits and Systems, Aug. 11-12, St. Louis, Mo. Contact R. Eugene Stuffle, Dept. of Electrical Engineering, 317 Engineering Research Laboratory, University of Missouri, Rolla, MO 65401-0249; (314) 341-4371.

Joint Fifth Symposium and Fifth International Conference on Logic Programming (Assoc. for Logic Programming), Aug. 15-19, Seattle. Contact Kenneth A. Bowen, LP88, Logic Programming Research Group, School of Computer and Information Science, 313 Link Hall, Syracuse University, Syracuse, NY 13210; (315) 423-2466.

ICPP 17, 17th International Conference on Parallel Processing, Aug. 15-19, St. Charles, Ill. Contact David H. Bailey, MS 258-5, NASA Ames Research Center, Moffett Field, CA 94035; Howard E. Sturgis, Xerox Corp., Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto, CA 94304; or Faye A. Briggs, Sun Microsystems, Inc., 2550 Garcia Ave., Mountain View, CA 94043.

DIAC 88, Directions and Implications of Advanced Computing Symposium (CPSR), Aug. 21, St. Paul, Minn. Contact Doug Schuler, Computer Professionals for Social Responsibility, PO Box 717, Palo Alto, CA 94301; (206) 865-3226.

Crypto 88 (IACR), Aug. 21-25, Santa Barbara, Calif. Contact Harold Fredricksen, Mathemetics Dept., Code 53Fs, Naval Postgraduate School, Monterey, CA 93943; (408) 646-2206.

CAD/CAM Technology Transfer to Latin America Conference (IFIP), Aug. 22-26, Mexico City. Contact Instituto de Investigaciones Electricas, Apdo. Postal 5-849, 11590 Mexico, D.F.

Coling 88, 12th International Conference on Computational Linguistics, Aug. 22-27, Budapest, Hungary. Contact Coling 88 Secretariat, Mtesz Congress Bureau, H-1055 Budapest, Kossuth ter. 6-8, Hungary.

LFA 88, Sixth IEEE Workshop on Languages for Automation, Aug. 29-31, College Park, Md. Contact Panos A. Ligomenides, Electrical Engineering Dept., University of Maryland, College Park, MD 20742; (301) 454-6842.

14th International Conference on Very Large Databases (VLDB Endowment), Aug. 29-Sept. 1, Los Angeles. Contact Jack E. Shemer, Teradata Corp., 12945 Jefferson Blvd., Los Angeles, CA 90066; (213) 827-8777.

EuroMicro 88, 14th Symposium on Microprocessing and Microprogramming, Aug. 29-Sept. 1, Zurich. Contact EuroMicro 88, PO 545, 7500 AM Enschede, The Netherlands; phone (31) 53-338-799.

ICO Topical Meeting on Optical Computing, Aug. 30-Sept. 2, Orsay, France. Contact S. Lowenthal, Insitut D'Uptique, BP 43, 91406 Orsay Cedex, France.

ITC 88, 1988 International Test Conference, Aug. 30-Sept. 2, Washington, DC. Contact ITC 88, Computer Society of the IEEE, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903; (202) 371-1013.

September 1988

NCGA Mapping and Geographic Information Systems 88, Sept. 12-15, Orlando, Fla. Contact NCGA, National Computer Graphics Assoc., 2722 Merrilee Dr., Suite 200, Fairfax, VA 22031.

Eurographics 88, Ninth European Assoc. for Computer Graphics Conference (INRIA), Sept. 12-16, Nice, France. Contact INRIA, Service des Relations Exterieures, Domaine de Voluceau, BP 105, 78153 Le Chesnay Cedex, France; phone (33) 1-39-635-501.

IEEE Artificial Neural Networks Conference, Sept. 18-21, Reston, Va. Contact Kamal Karma, 823 Flegler Rd., Gaithersburg, MD 20879; (301) 984-7657.

Fourth International Symposium on Biological and Artificial Intelligence Systems, Sept. 18-22, Trento, Italy. Contact Alberta Martino, IBM Corp., Dept. 48B/428, Neighborhood Rd., Kingston, NY 12401; (914) 385-4964.

Conference on Computerized Assistance During System Life Cycle (IFIP), Sept. 19-22, London. Contact International Federation for Information Processing, 3 rue de Marche, CH-1204, Geneva, Switzerland.

Workshop and Symposium on Formal Techniques in Real-time and Fault-tolerant Systems, Sept. 20-23, Coventry, England, UK. Contact M. Joseph, Dept. of Computer Science, University of Warwick, Coventry, CV4 7AL, UK.

First International Workshop on Transaction Machine Architecture, Sept. 25-28, Lake Arrowhead, Calif. Contact Martin Freeman, Signetics Corp., 811 E. Arques Ave., Sunnyvale, CA 94086; (408) 991-3591.

OOPSLA 88, Conference on Object-Oriented Programming: Systems, Languages, and Applications (ACM), Sept. 25-29, San Diego. Contact Barbara Noparstak, Digitalk, Inc., 9841 Airport Blvd., Los Angeles, CA 90045; (213) 645-1082.

Computational Intelligence 88, Sept. 26-30, Milano, Italy. Contact Giorgio Valle, Universita di Milano, Dip di Sciencze dell'Informazione, Via Moretto 9, 20133 Milano, Italy; phone (39) 02-2772-228.

26th Allerton Conference on Communication, Control, and Computing, Sept. 28-30, Monticello, Ill. Contact Allerton Conference, c/o Mark W. Spong, University of Illinois/Urbana-Champaign, Coordinated Science Laboratory, 1101 W. Springfield Ave., Urbana, Ill. 61801; (217) 333-4281.

October 1988

ICCD 88, IEEE International Conference on Computer Design: VLSI in Computers and Processors, Oct. 2-6, Rye Brook, N.Y. Contact ICCD 88, Computer Society of the IEEE, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903; (202) 371-1013.

EWSL 88, Third European Working Session on Learning, Oct. 3-5, Glasgow, Scotland. Contact Derek Sleeman, Dept. of Computing Science, University of Aberdeen, Aberdeen AB9 2UB, Scotland, UK; phone (44) 224-272-288.

Compsac 88, 12th International Computer Software and Applications Conference, Oct. 3-7, Chicago. Contact Wing N. Toy, Compsac 88, AT&T Bell Laboratories, Rm. 1Z-306, 200 Park Plaza, Naperville, IL 60566; (312) 416-4046.

1988 International Display Research Conference (IEEE, SID), Oct. 4-6, San Diego, Calif. Contact Palisades Institute for Research Services, Inc., Attn. IDRC, 201 Varick St., New York, NY 10014; (212) 620-3388.

International Workshop on Defect and Fault Tolerance in VLSI Systems, Oct. 6-7, Springfield, Mass. Contact Israel Koren, Dept. of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003; (413) 545-2643.

NCGA CAD/CAM 88, **Oct. 9-12,** Boston. Contact NCGA CAD/CAM 88, National Computer Graphics Assoc., 2722 Merrilee Dr., Suite 200, Fairfax, VA 22031.

ICCL 88, **International Conference on Computer Languages, Oct. 9-13,** Miami Beach, Fla. Contact Pei Hsia, Computer Science, University of Texas, Arlington, 2100 Oak Bluff Dr., Arlington, TX 76001; (817) 273-3785.

Seventh Symposium on Reliable Distributed Systems, Oct. 10-12, Columbus, Ohio. Contact Ming T. Liu, Dept. of Computer and Information Science, Ohio State University, 2036 Neil Ave., Columbus, OH 43210-1277; (614) 292-1837.

1988 IEEE Workshop on Visual Languages, Oct. 10-12, Pittsburgh. Contact Alfs T. Berztiss, Computer Science Dept., Faculty of Arts and Sciences, University of Pittsburgh, 322 Alumni Hall, Pittsburgh, PA 15260; or Stefano Levialdi, Departimento di Matematica, Universita di Roma, Piazzale A. Moro, 00185, Rome, Italy.

13th Conference on Local Computer Networks, Oct. 10-12, Minneapolis, Minn. Contact Ron Rutledge, MS-271, Martin Marietta Energy Systems, Oak Ridge National Laboratory, Oak Ridge, TN 37831; (615) 576-7643.

Frontiers 88, Second Symposium on the Frontiers of Massively Parallel Computation, Oct. 10-12, Fairfax, Va. Contact Frontiers 88 Symposium, PO 334, Greenbelt, MD 20770-0334.

International Workshop on Computer Vision (IAPR), Oct. 12-14, Tokyo. Contact Mikio Takagi, Institute of Industrial Science, University of Tokyo, 7-22-1, Roppongi, Minato-ku, Tokyo 106, Japan; phone (81) 3-479-0289.

ACM SIGGRAPH Symposium on **User Interface Software, Oct. 17-19,** Banff, Canada. Contact John Sibert, Dept. of Electrical Engineering and Computer Science, George Washington University, Washington, DC 20052; (202) 994-4953.

Fourth International Conference on Satellite Systems for Mobile Communications and Navigation (IEE), Oct. 17-19, London. Contact Conference Services Dept., Institution of Electrical Engineers, Savoy Pl., London WC2R 0BL, UK; phone (44) 01-240-1871.

Ninth International Conference on Pattern Recognition (IAPR), Oct. 17-20, Beijing. Contact Secretariat, Chinese Assoc. of Automation, PO Box 2728, Beijing, China.

ESIG 88, Fourth Expert Systems in Government Conference, Oct. 17-21, Washington, DC. Contact ESIG 88, Computer Society of the IEEE, 1730 Massachusetts Ave. NW, 20036-1903; (202) 371-1013.

Third International Symposium on Knowledge Engineering (AAAI), Oct.17-21, Madrid, Spain. Contact Jose R. Chelala, Third International Symposium on Knowledge Engineering, Alvarez de Baena, 3-2. °., 28006 Madrid, Spain.

Fourth Software Engineering Conference and Exhibition (AFCET), Oct. 18-21, Contact CGL4, 156, boulevard Pereire, 75017 Paris, France.

1988 Concom, Advances in Communications and Control Systems, Oct. 19-20, Baton Rouge, La. Contact Morteza Naraghi-Pour, Dept. of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803; (504) 388-5551.

CSM 88, Conference on Software Maintenance—1988 (DPMA, NBS), Oct. 24-27, Phoenix, Ariz. Contact Robert Arnold, Software Productivity Consortium, 1880 Campus Commons Dr. North, Reston, VA 22091; (703) 391-1898.

Second International Software for Strategic Systems Conference, Oct. 25-26, 1988, Huntsville, Ala. Contact Karen B. Mack, University of Alabama, Division of Continuing Education, Tom Bevill Center, Huntsville, AL 35899; (205) 895-6357.

JOIN THE TECHNICAL COMMITTEE ON MULTIPLE-VALUED LOGIC

The Multiple-Valued Logic TC promotes research in the theory and application of multiple-valued systems. Its scope includes: devices, VLSI, circuit design techniques, fault tolerance, probabilistic and variable-valued systems, optical computing, fuzzy logic, automated reasoning, and philosophical aspects.

DID YOU KNOW THAT

The International Symposium on Multiple-Valued Logic (ISMVL) is sponsored each year by the MVL-TC with proceedings published by the Computer Society Press. Membership includes researchers all over the world: North and South America, Europe, and Asia. There are also local MVL TCs in Japan and China. The next ISMVL will be held May 24-26, 1988, in Palma de Mallorca, Spain. Future sites include Charlotte, NC, and Sendai, Japan.

RECENT DEVELOPMENTS IN MULTIPLE-VALUED LOGIC

Significant applications of MVL techniques include implementation and use of four-valued ROMs by Intel, Motorola, and General Instrument; MVL techniques for optimization of binary PLAs; applications of fuzzy logic to database systems; a high speed VLSI multiplier; and a fuzzy logic inference chip.

Please Tear below Line

FOR MORE INFORMATION PLEASE CHECK THE BOXES BELOW
THAT REFLECT YOUR INTERESTS:

( ) Information on the International Symposium
( ) Information about Technical Committee Publications
( ) Information about Membership in the Technical Committee

YOUR NAME:

YOUR ADDRESS:

RETURN THE BOTTOM PART OF THIS ADVERTISEMENT TO:
Dr. David Rine, Technical Committee on MVL
Computer Science Department
George Mason University
Fairfax, Virginia USA 22030

MVL TECHNICAL COMMITTEE

THE COMPUTER SOCIETY ®

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. ®

IECON 88 (IEEE), Oct. 25-27, Singapore. Contact V.K.L. Huang, AT&T Information Systems, Crawford Corners Rd., Holmdel, NJ 07733; (210) 949-0069.

Ninth IEEE Symposium on Mass Storage Systems, Oct. 30-Nov. 3, Monterey, Calif. Contact Patric Savage, Shell Development Co., PO 481, Houston, TX 77001; (713) 663-2384.

November 1988

ICCAD 88, IEEE International Conference on Computer-Aided Design (ACM) Nov. 7-10, Santa Clara, Calif. Contact Al Jimenez, Mentor Graphics Corp., 1940 Zanker Rd., San Jose, CA 95112; (408) 436-1500.

Second International Symposium on Interoperable Information Systems, Nov. 10-11, Tokyo. Contact Hideo Aiso, Dept. of Electrical Engineering, Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Karagawa, 223 Japan; (81) 044-631-141, ext. 3320.

Supercomputing 88, Nov. 14-18, Orlando, Fla. Contact George Michael, L-306, Lawrence Livermore National Laboratory, PO 808, Livermore, CA 94550; (415) 422-4239.

Late Magazines?
No Magazines?
Membership Status Problems?
No Answers To Your Complaints?
Let your Computer Society Ombudsman cut through the red tape for you.

Computer Society Ombudsman
IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

CALL FOR PAPERS

Call for papers for *Computer*

Computer seeks articles for the special January 1989 theme issue on **Real Machines: Design Choices/Engineering Tradeoffs.**

Six copies of each complete manuscript should be submitted by **May 15, 1988** to Yale Patt, *Computer* Guest Editor, Computer Science Division, Dept. of Electrical Engineering and Computer Science, 573 Evans Hall, University of California, Berkeley, CA 94720.

1988 IEEE Workshop on Visual Languages: Oct. 10-12, 1988, Pittsburgh. Submit paper by **Apr. 15, 1988** to Alfs T. Berztiss, Computer Science Dept., Faculty of Arts and Sciences, University of Pittsburgh, 322 Alumni Hall, Pittsburgh, PA 15260; or (in Europe) to Stefano Levialdi, Dipartimento di Matematica, Universita di Roma, Piazzale A. Moro, 00185, Rome, Italy.

Frontiers 88, Second Symposium on the Frontiers of Massively Parallel Computation: Oct. 10-12, 1988, Fairfax, Va. Submit abstract by **Apr. 15, 1988** to David H. Schaefer, Dept. of Electrical and Computer Engineering, George Mason University, 4400 University Dr., Fairfax, VA 22030.

SIGSoft 88, Third Symposium on Software Development Environments (ACM): Dec. 5-7, 1988, Boston. Submit paper by **Apr. 15, 1988** to Barry Boehm, Chair SDE, TRW-DSG, One Space Park, R2-2086, Redondo Beach CA 90278.

12th Western Educational Computing Conference (California Educational Computing Consortium): Nov. 17-18, 1988, San Diego, Calif. Submit paper by **Apr. 15, 1988** to Oliver Seely, California State University, Dominguez Hills, 1000 E. Victoria St., Carson, CA 90747; (213) 516-3383.

Globecom 88, IEEE Global Telecommunications Conference: Nov. 28-Dec. 1, 1988, Hollywood, Fla. Submit complete manuscript by **Apr. 18, 1988** to Ray R. Laane, Bell Communications Research, 435 South St., Rm. 2F-287, Morristown, NJ 07960; (201) 829-4064.

ICCAD 88, IEEE International Conference on Computer-Aided Design (ACM): Nov. 7-10, 1988, Santa Clara, Calif. Submit paper by **Apr. 29, 1988** to ICCAD 88 Secretary, Electrical and Computer Engineering Dept., Carnegie Mellon University, Pittsburgh, PA 15213; (412) 268-3546.

Sixth CIPS Edmonton Computer Conference: Nov. 15-17, 1988, Edmonton, Canada. Submit paper by **Apr. 30, 1988** to U.M. Maydell, Dept. of Computing Science, University of Alberta, Edmonton, Alberta, T6G 2H1, Canada.

13th Conference on Local Computer Networks: Oct. 10-12, 1988, Minneapolis, Minn. Submit paper by **May 1, 1988** to Larry Green, Advanced Computer Communication, 720 Santa Barbara St., Santa Barbara, CA 93101; (805) 963-9431.

Ninth Real-Time Systems Symposium: Dec. 6-8, 1988, Huntsville, Ala. Submit paper by **May 1, 1988** to Al K. Mok, Dept. of Computer Science, TAY 3-140C, University of Texas/Austin, Austin, TX 78712; (512) 471-9542.

International Symposium on Databases in Parallel and Distributed Systems: Dec. 5-7, 1988, Austin, Texas. Submit manuscript by **May 1, 1988** to Won Kim, MCC, 3500 Balcones Center Dr., Austin, TX 78759; (512) 338-3439.

International Seminar on Performance of Distributed and Parallel Systems: Dec. 7-9, 1988, Kyoto, Japan. Submit paper by **May 1, 1988** to Yutaka Takahashi, Dept. of Applied Mathematics and Physics, Kyoto University, Kyoto 606, Japan; (81) 075-751-2111.

Impact of Recent Computer Advances on Operations Research (ORSA): Jan. 4-6,

Conferences that the Computer Society sponsors or participates in are indicated by the Computer Society logo; additional conference sponsors are listed in parentheses. Other conferences of interest to our readers are also included.

For inclusion in Call for Papers or Calendar, submit information six weeks before the month of publication (i.e., for the **June 1988** issue, send information for receipt by **April 15, 1988**) to Calendar Editor, *Computer*, 10662 Los Vaqueros Cir., Los Alamitos, CA 90720.

1989, Williamsburg, Va. Submit paper by **May 1, 1988** to Ramesh Sharda, College of Business Administration, Oklahoma State University, Stillwater, OK 74078; (405) 624-5113.

EWSL 88, Third European Working Session on Learning: Oct. 3-5, 1988, Glasgow, Scotland. Submit paper by **May 1, 1988** Derek Sleeman, Dept. of Computing Science, University of Aberdeen, Aberdeen AB9 2UB, Scotland, UK; phone (44) 224-272-288.

1988 Workshop on VLSI Signal Processing: Nov. 2-4, 1988, Monterey, Calif. Submit extended summary by **May 1, 1988** to Paulette Powell, Electronics Research Laboratory, University of California, Berkeley, CA 94720; (415) 642-1566.

1988 International Display Research Conference (IEEE, SID): Oct. 4-6, 1988, San Diego, Calif. Submit abstract by **May 2, 1988** to Palisades Institute for Research Services, Inc., Attn. IDRC, 201 Varick St., New York, NY 10014; (212) 620-3388.

Fourth Conference on Artificial Intelligence for Space Applications (NASA): Nov. 15-16, 1988, Huntsville, Ala. Submit abstract by **May 2, 1988** to Conference on Artificial Intelligence for Space Applications, NASA/EB44, MSFC, AL 35812; (205) 544-5181.

FGCS 88, International Conference on Fifth Generation Computer Systems: Nov. 28-Dec. 2, 1988, Tokyo. Submit manuscript by **May 10, 1988** to Hidehiko Tanaka, Institute for New Generation Computer Technology, Mita Kokusai Bldg. 21F, 1-4-28 Mita, Mianto-ku, Tokyo 108, Japan.

ICCV 88, Second International Conference on Computer Vision: Dec. 5-8, 1988, Tarpon Springs, Fla. Submit paper by **May 15, 1988** to Ruzena Bajcsy, University of Pennsylvania, Dept. of Computer and Information Science, 200 S. 33rd St., Philadelphia, PA 19104-6389; (215) 898-6222.

International Workshop on Computer Vision (IAPR): Oct. 12-14, 1988, Tokyo. Submit summary by **May 15, 1988** and final paper by **Aug. 31, 1988** to Mikio Takagi, Institute of Industrial Science, University of Tokyo, 7-22-1, Roppongi, Minato-ku, Tokyo 106, Japan; phone (81) 3-479-0289.

Third Workshop on Knowledge Acquisition for Knowledge-Based Systems (AAAI): Nov. 7-11, 1988, Banff, Canada. Submit abstract or complete paper by **May 15, 1988** to John H. Boose, Advanced Technology Center, Boeing Computer Services, MS 7L-64, PO 24346, Seattle, WA 98124; (206) 865-3253.

Fourth National Convention of Computer Engineers (Institution of Engineers—India): December 1988, Calcutta, India. Submit paper by **May 15, 1988** to Anirban Basu, Fourth National Convention of Computer Engineers, West Bengal State Center, 8, Gokhale Rd., Calcutta-700 020, India.

Eighth Conference on Foundations of Software Technology and Theoretical Computer Science: Dec. 21-23, 1988, Pune, India. Submit paper by **May 16, 1988** K.V. Nori, TRDDC, 1 Mangaldas Rd., Pune 411001, India; phone (212) 61608.

Fourth Aerospace Computer Security Applications Conference: Dec. 12-16, 1988, Orlando, Fla. Submit paper by **May 20, 1988** to William T. Bisignani, Booz-Allen & Hamilton, Inc. 4330 East-West Hwy., Bethesda, MD 20814.

Symposium on Computer Graphics Education: Nov. 4-5, 1988, Poughkeepsie, N.Y. Submit paper by **May 31, 1988** to Marek Holynski, Boston University, 111 Cummington St., Boston, MA 02215.

IEEE Software: January 1989. Articles are sought on human-computer interfaces, human factors, and user issues. Submit articles by **June 1, 1988** to Virginia Hix, Guest Editor, Computer Science Dept., Virginia Polytechnic State University, Blacksburg, VA 24061; (703) 961-4857, or Ted G. Lewis, Editor-in-Chief, *IEEE Software* c/o Computer Science Dept., Oregon State University, Corvallis, OR 97331; (503) 754-2744; CSnet lewis@oregon-state; Compmail + t.lewis.

1988 International Computer Science Conference: Dec. 19-21, 1988, Hong Kong. Submit paper by **June 15, 1988** to Jean-Louis Lassez, IBM T.J. Watson Research Center, PO 218, Yorktown Heights, NY 10598, or Francis Y.L. Chin, Center of Computer Studies and Application, University of Hong Kong, Hong Kong.

Fifth International Conference on Data Engineering: Feb. 7-9, 1989, Los Angeles. Submit paper by **June 15, 1988** to Richard L. Shuey, Computer Science Dept., Rensselaer Polytechnic Institute, Troy, NY 12189-3590; (518) 276-8376 or (518) 374-5684.

IFIP TC-2 Working Conference on Visual Database Systems (IPSJ): Apr. 3-7, 1989, Tokyo. Submit paper by **June 15, 1988** to IFIP TC-2 Working Conference, Tosiyasu L. Kunii, Dept. of Information Science, Faculty of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan; phone (81) 03-812-2111, ext. 4116.

Proceedings of the IEEE: January 1990. A special issue on state-of-the-art surveys on supercomputing technology is planned. Submit paper or proposal by **June 30, 1988** to Tse-yun Feng or A.R. Hurson, E.E. East Bldg. Pennsylvania State University, University Park, PA 16802; (814) 863-1469 or 1187.

IEEE Transactions on Computers (IEEETC): April 1989. Papers are sought for a special issue on high-yield VLSI systems. Submit manuscript by **July 1, 1988** to Israel Koren, Dept. of Electrical and Computer Engineering, 210 Marcus Hall, University of Massachusetts, Amherst, MA 01003; (413) 545-2643.

Second International Workshop on Artificial Intelligence in Economics and Management (IFIP, IFAC, IFORS): Jan. 11-13, 1989, Singapore. Submit extended abstract by **July 1, 1988** to Vicky Toh, Institute of Systems Science, National University of Singapore, Kent Ridge, Singapore 0511.

26th Allerton Conference on Communication, Control, and Computing: Sept. 28-30, 1988, Monticello, Ill. Submit paper and extended abstract by **July 15, 1988** to Allerton Conference, c/o Mark W. Spong, University of Illinois/Urbana-Champaign, Coordinated Science Laboratory, 1101 W. Springfield Ave., Urbana, Ill. 61801; (217) 333-4281.

ICS 88, 1988 International Computer Symposium: Dec. 20-21, 1988, Taipei, Taiwan. Submit paper by **July 15, 1988** to Louis R. Chow, Tamkang University, Tamsui, Taiwan, Republic of China (for authors in the Far East) or to Shi-Kuo Chang, Dept. of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260 (for authors outside the Far East).

IEEE Software: Articles are sought for a special issue on software technology in the Far East. Submit manuscripts by **Aug. 1, 1988** to Carl Chang, Guest Editor, EECS Dept., MC 154, University of Illinois, PO 4348, Chicago, IL 60680; (312) 996-4860.

IEEE Infocom 89, Conference on Computer Communications: Apr. 24-27, 1989, Ottawa, Canada. Submit paper by **Aug. 1, 1988** to J.W. Mark, IEEE Infocom 89, Dept. of Electrical Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1; (519) 888-4016.

CompEuro 89, International Computer Conference: May 8-12, 1989, Hamburg. Submit paper by **Aug. 15, 1988** to Walter E. Proebster, IBM Laboratory, PO Box 1380, D-7030 Boeblinger, Federal Republic of Germany.

CAPE 89, Computer Applications in Production Engineering Conference (IFIP, IPSJ, JSPE): Oct. 2-5, 1989, Tokyo. Submit abstract by **Sept. 30, 1988** to Business Center for Academic Societies Japan, 2-40-14, Hongo, Bunkyo-ku, Tokyo 113, Japan; phone 001 (81) 3-817-5831.

IFIP Congress 89, 11th World Computer Congress: Aug. 28-Sept. 1, 1989, San Francisco. Submit paper by **Nov. 1, 1988** to Herve Gallaire, ECRC, Arabellastrasse 17, D-8000 Munich 81, Federal Republic of Germany; phone (49) 89-9269-9100.

MIV 89, International Workshop on Industrial Applications of Machine Intelligence and Vision IEEE): Apr. 10-12, 1989, Tokyo. Submit summary by **Nov. 21, 1988** to Mitsuru Ishizuka, Institute of Industrial Science, University of Tokyo, 7-22-1, Roppongi, Minato-ku, Tokyo, 106, Japan; phone 81 (03) 470-5389.

CAREER OPPORTUNITIES

RATES: $12.00 per line, $120 minimum charge (up to ten lines). Average six typeset words per line, nine lines per column inch. Add $10 for box number. Send copy at least one month prior to publication to: **Heidi Rex or Marian Tibayan, Classified Advertising, COMPUTER Magazine, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720.**

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, *COMPUTER* may reject any advertisement containing any of these phrases or similar ones: "...recent college grads...," "...1-4 years maximum experience...," "...up to 5 years experience...," or "...10 years maximum experience." *COMPUTER* reserves the right to append to any advertisement, without specific notice to the advertiser, "Experience ranges are suggested minimum requirements, not maximums." *COMPUTER* assumes that, since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only.

McGILL UNIVERSITY
School of Computer Science

Applications are invited for a Full Professorship in distributed computing with special interest in Software Design. The position is for a prestigious Chair funded by Bell Northern Research (BNR) and Northern Telecom and is open to Full Professors, senior Associate Professors with an excellent publication record in the field or to people with an equivalent industrial qualification. Some research areas of interest are: interprocessor communication methods, distributed operating systems software, design of distributed name services, fault tolerance in distributed systems, distributed file systems, distributed data bases and transaction processing, distributed algorithms, debugging in distributed systems, programming languages for distributed applications, performance analysis of distributed systems as well as campus, metropolitan and wide area distributed systems.

The School of Computer Science currently consists of 17 faculty and offers B.Sc., M.Sc. and Ph.D. programs. A new major program in Computer Architecture and Programming is being established. The possibility exists for the Full Professor to build a small team with the cooperation of a government-industry program. The Centre de Recherche en Informatique de Montreal offers unique opportunities for university professors to carry on applied research in cooperation with the leading Canadian industries.

In accordance with Canadian immigration requirements, priority will be given to Canadian citizens and permanent residents of Canada.

Applicants should submit a curriculum vitae and the names of four referees to Professor Pierre Belanger, Dean, Faculty of Engineering, McGill University, 817 Sherbrooke Street West, Montreal, Quebec, CANADA H3A 2K6.

THE PENNSYLVANIA STATE UNIVERSITY
Computer Engineering

Applications are invited for tenure-track and visiting faculty positions at all levels. Candidates from all areas of computer engineering (hardware and software) will be considered. The Computer Engineering Program at The Pennsylvania State University is within the Department of Electrical Engineering which has over 50 faculty members, and approximately 1500 undergraduate majors, 170 graduate students. Candidates should have a Ph.D. in Electrical/Computer Engineering or related areas. There are 13 faculty members within the Computer Engineering Program. Excellent instruction and research computing facilities are available within the Department, College and at the University Computing Center.

Please send your letter of application, resume, or inquiries, together with three references to: T. Feng, Computer Engineering Program, Department of Electrical Engineering 129 Electrical Engineering East, Box ES, The Pennsylvania State University, University Park, PA 16802.

Deadline for applications is June 30, 1988 or until suitable qualified candidates are selected. "An Equal Opportunity/Affirmative Action Employer."

SENIOR RESEARCH COMPUTER SCIENTIST

Senior Research Computer Scientist—research and Ph.D. supervision in areas of computer vision and robotics; Synthetic Aperture Radar Image Understanding System to generate special high speed recognition programs for objects such as tanks and airplanes in SAR images; develop computer systems to process and extract information for images including new geometric modeler for SAR Image Understanding System and hand-eye system; REQUIRES: Ph.D. information engineering plus five years experience and state-of-the-art research experience in areas related to job; must have hands on experience in photometric stereo, light-stripe range finder and PUMA arm, geometric modeler, GEOMAP, geometric modeler PADL-11, commonlisp, framekit language, and C; strong academic research and publication record. Send resume or appear at Job Service East, 6206 Broad Street, Pittsburgh, PA 15206. J.O. #3849840.

SOUTHWESTERN OKLAHOMA STATE UNIVERSITY

Computer Science: Southwestern Oklahoma State University invites applications for a faculty position in Computer Science. Requirement: Ph.D. or M.S. degree in Computer Science or closely related field (Ph.D. preferred). Primary faculty responsibility is teaching introductory and upper level undergraduate Computer Science courses, but research is encouraged. Salary is commensurate with qualifications: mid 30's for nine-month contract with Ph.D. Tenure track position with Ph.D. beginning June or August, 1988. Review of applications will begin April 4. Send resume and list of professional references with cover letter to: Personnel Office, Southwestern Oklahoma State University, 100 Campus Drive, Weatherford, OK 73096. SWOSU is an equal opportunity, affirmative action employer.

UNIVERSITY OF ALBERTA
Department of Computing Science

Applications are invited for two tenure-track positions at the Assistant/Associate Professor level. Responsibilities include research as well as teaching at the graduate and undergraduate levels. Strong candidates from all research areas will be considered, but areas of special interests include database systems, VLSI/computer architecture, operating systems, numerical analysis and computer graphics.

The Department consists of 36 academic and 28 support staff. Current hardware support includes an Amdahl 5870, a network of four VAX 11/780's and about thirty Sun Workstations, and well-equipped microcomputer and workstation laboratories for graphics, VLSI, and AI research. Access to a Cyber 205 is available.

Salary range $32,756 to $57,236 and is commensurate with qualifications and experience. Send curriculum vitae and the names of three references, and up to three reprints or copies of important publications. New Ph.D.'s should also include a copy of their transcript. Apply to:

Dr. Lee J. White, Chairman
Department of Computing Science
University of Alberta
Edmonton, Alberta. Canada
T6G 2H1

Applications will be accepted until June 30, 1988.

The University of Alberta is committed to the principle of equity in employment

THE HARTFORD GRADUATE CENTER
Affiliated with Rensselaer Polytechnic Institute
Department Chair—Computer Science
Search Reopened

The Hartford Graduate Center is an independent institution accredited to award Master's degrees from Rensselaer Polytechnic Institute of Troy, New York as well as in its own right. The Graduate Center invites applications from experienced persons for the position of Chairperson, Department of Computer Science, for the Fall 1988 semester.

Research interests within the Department include Computer Vision, Human-Computer Interaction, Software Engineering, Operating Systems, Computer Engineering and Database. The Department maintains a superb workstation-based computing facility (Sun 3 and Apollo) to support these interests and the teaching program. A Ph.D. in Computer Science or a related area is required. Preference will be given to those applicants with a background in Computer Communications Networks, Software Engineering, or Artificial Intelligence. Closing date for applications is May 15, 1988, and the projected starting date for the appointment is for the '88-'89 academic year.

Direct vita, along with three references to:

Michael M. Danchak
Dean, Engineering and Science
The Hartford Graduate Center
275 Windsor St.
Hartford, CT 06120-2991

The Hartford Graduate Center is an Equal Opportunity/Affirmative Action Employer.

YARMOUK UNIVERSITY
Department of Computer Science

The Department of Computer Science at Yarmouk University of Jordan invites applications for tenure-track faculty positions at the rank of Assistant, Associate and Professor in all areas of computer science.

We are especially interested in operating systems, compiler design, artificial intelligence, computer architecture networks, simulation, graphics, testing systems, office automation, programming languages design and implementation, numerical analysis, theoretical computer science.

Duties will include teaching graduate and undergraduate courses. Candidates should hold a Ph.D. in computer science or related discipline. Teaching responsibility are supported by a VAX 11/780, WANG VS-100 and several types of microcomputers.

Applications for visiting appointments are also encouraged. Interviews place and date will be communicated. In order to assure full consideration, application should be received by April 30, 1988. Interested persons should send a complete vitae and names of three references to: Academic Staff Affairs Division—Yarmouk University—IRBID—JORDAN

CALIFORNIA STATE UNIVERSITY, FULLERTON

The Department of Electrical Engineering at California State University, Fullerton, seeks applications for tenure-track and lecturer positions starting in August 1988. Rank and salary commensurate with qualifications. Applicants for tenure-track must hold a Ph.D. or D.Sc. in Electrical Engineering and/or Computer Engineering with specialization in computers, communications and signal processing, systems, robotics. VLSI system design or related fields. California State University, Fullerton is a major urban university in Orange County, located close to many major high-technology corporations whose levels of sophistication in engineering are world renowned. As a result of this environment, the department currently serves over 1300 majors at the B.S. and M.S. levels. The department has curricular strength in the areas of communications, signal processing, digital systems, control systems, systems engineering, solid state electronics and electromagnetism. CSUF stresses the importance and value of teaching and research excellence. Successful applicants must be motivated to that end, and have the desire and ability to clearly communicate in the classroom/laboratory environment. Research is strongly emphasized and is an important element of a faculty member's overall scholarly and creative activity. A new engineering building addition, essentially doubling the classroom, office and laboratory capacity of engineering, is expected to be occupied in the Fall of 1989. Plans are to include state-of-the-art laboratories in robotics, VLSI design, digital control systems, microwave and computer aided design. Candidates must indicate interest in tenure-track or lecturer positions or both. Minorities and women are particularly encouraged to consider joining this dynamic department. Send resume and three references to Dr. George I. Cohn, Chair, Search Committee, Electrical Engineering Department, California State University, Fullerton, CA 92634. Deadline for tenure track applicants is April 15, 1988 however, positions are normally opened until filled. Lecturer applicants are accepted until all positions are filled. CSUF is an Affirmative Action/Equal Opportunity/Title IX employer.

RENSSELAER POLYTECHNIC INSTITUTE
Faculty Positions
Department of Electrical, Computer, and Systems Engineering

Applications are invited for tenure-track faculty positions at all levels. Areas of interest include but are not limited to: computer engineering, architecture and parallel processing, hardware design, and performance evaluation, computer networks, VLSI design, optical computing and communications, knowledge-based systems and artificial intelligence. The ECSE Department is the largest academic unit at RPI and has a rich tradition of research and education. The department is seeking to add faculty who bring an innovative approach to research and teaching. Active programs in computer engineering, solid-state electronics and integrated circuit design, control systems, robotics and automation, information and decision systems, communications and signal processing, electronics and circuits, and fusion plasma systems contribute to a dynamic research environment. In addition to the extensive research facilities of the department, there are opportunities to initiate or participate in interdisciplinary research programs in one of the major research centers of the School of Engineering, including the Center for Integrated Electronics, Center for Interactive Computer Graphics, and Center for Manufacturing Productivity and Technology Transfer. New faculty are eligible for special arrangements including summer support, equipment, graduate student support, and reduced teaching load in order to encourage growth of their research programs. Applications or requests for more information should be directed to:

Dr. Arthur C. Sanderson
Chairman, Department of Electrical,
Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

RPI is an affirmative action/equal opportunity employer.

THE UNIVERSITY OF TENNESSEE
Department of Computer Science
Knoxville, Tennessee 37996-1301

The Department of Computer Science invites applications for tenure-track and visiting positions at all ranks beginning Winter of 1988. A doctoral degree in computer science or a related area, and a strong interest in research in the areas of computer networks, computer graphics, and operating systems are required. Those with experience directing doctoral students are especially encouraged to apply.

The department operates a research laboratory which contains two DEC VAX-8200's (one VMS, one UNIX) and a variety of microcomputers. Each faculty member has a personal workstation. Specialized laboratories support research in areas of special emphasis. The University of Tennessee Computing Center operates an IBM 3081, an IBM 4381, and a cluster of VAX 8600's. Faculty members collaborate with scientists at the Oak Ridge National Laboratory and frequently have access to their facilities, including a Cray X-MP and hypercube computers, under the terms of the collaboration. Faculty workstations are networked with the departmental laboratories, the University Computing Center, and facilities at the Oak Ridge National Laboratory.

You can respond on CSNET with csnet:straight @utenn and on BITNET with bitnet:straight@ utkcs1. The mailing address is Computer Science Department, 107 Ayres Hall, The University of Tennessee, Knoxville TN 37996-1301.

The University of Tennessee is an EEO/TITLE IX/SECTION 504 employer.

SHAPE TECHNICAL CENTRE
THE HAGUE, THE NETHERLANDS

This NATO Scientific and Technical establishment has vacancies for Information systems scientists. The work centers around the definition of future large scale, advanced Military Command and Control Information Systems in age where the preferred approach is evolutionary development, combining information systems analysis with the development of prototypes in the centre's laboratories. Candidates should have in-depth experience in one or more of the following areas: Interoperability of heterogeneous systems, ISO/OSI fourth generation tools and rapid prototyping.

Applicants should have a master's degree, preferably supplemented by a postgraduate qualification in computer science, several years' relevant postgraduate experience and the ability to communicate effectively at all levels in English, both orally and in writing. Shape Technical Centre offers an excellent salary, allowances and additional privileges for expatriate staff. There are generous annual and home leave arrangements and a good pension and private health insurance scheme. A three-year contract is offered which may be renewed by mutual consent.

Interested NATO Nationals are requested to forward curriculum vita (in English) to:

Personnel Officer, Shape Technical Centre, PO Box 174, 2501 CD The Hague, The Netherlands.

INDIANA UNIVERSITY OF PENNSYLVANIA
Finance/MIS Department

Of the more than 200 colleges and universities in Pennsylvania, Indiana University of Pennsylvania is the fifth largest with an enrollment of 13,405 students on its main campus and two branches. The University, employing over 1500 employees, including a faculty of approximately 700, consists of six colleges and two schools and offers more than 100 majors within 40 departments. Located just 50 miles Northeast of Pittsburgh, IUP is located in the foothills of the Allegheny Mountains in Indiana, PA a community of 35,000 residents.

INDIANA UNIVERSITY OF PENNSYLVANIA invites applications for Assistant/Associate Professor (tenure-track) to teach undergraduate or graduate courses in Management Information Systems.

Qualifications: Ph.D., DBA or ABD in MIS or related field, e.g., Information Sciences, Computer Science, or Information Systems from Industrial Engineering Disciplines. Experience in teaching, evidence of scholarly research in MIS or Information Technology, experience with micros and mainframes, and active in professional fields.

Duties: Appointment will be for nine months. All full-time positions are tenure eligible. Qualified candidate will teach undergraduate and graduate courses in the following areas: Data Base, Management Systems, Systems Analysis and Design, COBOL Programming, Telecommunications, and use mainframes and micros as needed. The candidate will also advise students; serve on Department, College and University committees; develop new courses; and do research in the candidate's area of specialty.

Send application, resume, transcripts and three references to: Dr. Ata Nahourail, Chair, Search Committee, Finance/MIS Department, Indiana University of Pennsylvania, Indiana, PA 15705. Telephone (412) 357-4740. Review of applications will begin January 5, 1988 and will continue until position is filled.

IUP is an Affirmative Action/Equal Opportunity Employer.

NAVAL POSTGRADUATE SCHOOL, Monterey, CA
Position Announcement in Computer Science

The Department of Computer Science has immediate openings for faculty positions at all levels. Our primary interests are in the areas of operating systems, programming languages, and algorithms. Our secondary interests are in the areas of processing of visual data, real-time systems, and software engineering. An applicant should have a Ph.D. in Computer Science or a related field and have a strong interest in both graduate teaching and research. Senior applicants must have distinguished research records. Appointments can begin at any time during the year.

The Department offers MS and Ph.D. degrees in Computer Science supported by well-equipped instructional/research facilities and full-time technical staff. The faculty normally teach for two quarters and conduct full-time research during the other two quarters.

Please send a detailed resume and three letters of reference to: Search Committee, Computer Science Department (Code 52), Naval Postgraduate School, Monterey, CA 93943, tel. #(408) 646-2449. NPS IS AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER.

NAVAL POSTGRADUATE SCHOOL
Position Announcement in Computer Science

The Department of Computer Science has immediate openings for faculty positions at all levels. Our primary interests are in the areas of operating systems, distributed systems, programming languages, and algorithms. Our secondary interests are in the areas of processing of visual data, real-time systems, and software engineering. An applicant should have a PhD in Computer Science or a related field and have a strong interest in both graduate teaching and research. Senior applicants must have distinguished research records. Appointments can begin at any time during the year.

The Department offers MS and PhD degrees in Computer Science. Departmental facilities (supported by 9 full-time computer professionals) consist of six instructional and research laboratories including the Database Systems Laboratory, the Graphics and Video Laboratory, the Software Engineering Laboratory, the Micro computer Systems Laboratory, the Artificial Intelligence Laboratory and the Computer Science Academic Laboratory. In these laboratories are the latest, state-of-the-art computing equipment including LISP machines, high-performance graphics workstations, bit-mapped graphics workstations, multi-microprocessor systems, etc. During the academic year, the faculty normally teach for two quarters and conduct full time research supported by major research programs in both the Department of Defense and the non-DOD organizations during the other two quarters. New faculty receive initial support from an on-campus research foundation.

Located on Monterey Bay, the areas of Monterey, Pebble Beach, and Carmel provide a pleasant Northern California coastal climate and easy access to Silicon Valley companies and universities.

Please send a detailed resume and three letters of reference to:

Search Committee
Computer Science Department (Code 52)
Naval Postgraduate School
Monterey, CA 93943
Tel. No. (408) 646-2449

AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER.

INVENTIONS WANTED OR MISC.

Ideas, inventions, new products wanted by one of America's leading invention submission firms to submit to industry and exhibit at National Inventor's Exposition. Call free 1-800-288-IDEA. Canada, 1-800-528-6060. Extension 831. Or write: ISC-COMP, 903 Liberty Avenue, Pittsburgh, PA 15222.

THE UNIVERSITY OF TENNESSEE AT CHATTANOOGA
Director—Center of Excellence for Computer Applications

The University of Tennessee at Chattanooga is accepting applications and nominations for the directorship of the Center of Excellence for Computer Applications (CECA). CECA has an annual budget of one and a half million dollars and twenty full time faculty and staff organized into five program areas (computer curricular integration, computer resources support, engineering applications, knowledge based applications and rehabilitation interfaces engineering). CECA is concerned with a broad spectrum of computer applications in engineering, business, arts and sciences, education, and health and human services. As a part of its mission, the Center supports an excellent academic computing facility with a number of microcomputer laboratories, two HP-3000 minicomputers, an IBM 4381, and remote access to the University of Tennessee Computer Center network which includes an IBM 3081-D, an IBM 3090-200 and three DEC VAX systems.

The successful candidate must hold an earned doctorate in an appropriate discipline and possess the qualifications for an academic appointment at the level of full professor, tenurable in the department representing the applicant's academic discipline. Applicants must have an established record of research and publication, possess significant experience in areas of interest to CECA, have extensive experience in securing and managing externally funded grants and contracts and have experience in administering a funded research unit. Leadership experience with a computer facility in an academic setting is highly desirable.

The appointment is on a twelve month basis. The salary is competitive and is dependent on the applicant's background and qualifications.

Applications should include a letter of interest, current resume and the names and addresses of three references. Nominations and application should be sent to:

Dr. Jack Thompson, Chair
CECA Director Advisory Search Committee
Center of Excellence For Computer Applications
413 Hunter Hall
The University of Tennessee At Chattanooga
Chattanooga, TN 37403

Applications will be reviewed beginning April 22, 1988. An appointment will be made as soon as the successful candidate is selected. The starting date for the successful candidate can be negotiated.

The University of Tennessee at Chattanooga offers a broad range of liberal arts and professional degrees. The University has approximately 7,400 students including 1,400 graduate students, 330 faculty, an annual budget of $50 million, and an endowment of more than $33 million. The Center of Excellence for Computer Applications is funded by special state appropriations as a part of Tennessee's Better Schools Program. UTC is located in southeastern Tennessee, one of the most scenic areas of the country. It is adjacent to mountain and lake recreation areas and is within a 2-hour drive to Atlanta, Nashville, and Birmingham.

The University of Tennessee at Chattanooga is an equal employment opportunity, affirmative action Title IX, Section 504 institution.

THE HARTFORD GRADUATE CENTER
Affiliated with Rensselaer Polytechnic Institute
Computer Science Faculty

The Hartford Graduate Center is an independent institution accredited to award Master's degrees from Rensselaer Polytechnic Institute of Troy, New York as well as in its own right. The Graduate Center invites applications for faculty positions, at all levels, in the Department of Computer Science, for the Fall 1988 semester.

Research interests within the Department include Computer Vision, Human-Computer Interaction, Software Engineering, Operating Systems, Computer Engineering, and Database. The Department maintains a superb workstation-based computing facility (Sun 3 and Apollo) to support these interests and the teaching program. A Ph.D. in Computer Science or a related area is required. Preference will be given to those applicants with a background in Computer Communications Networks, Software Engineering, or Artificial Intelligence. Closing date for applications is May 15, 1988, and the projected starting date for the appointment is for the '88-'89 academic year.

Direct vita, along with three references to:

Michael M. Danchak
Dean, Engineering and Science
The Hartford Graduate Center
275 Windsor St.
Hartford, CT 06120-2991

The Hartford Graduate Center is an Equal Opportunity/Affirmative Action Employer.

CASE WESTERN RESERVE UNIVERSITY
Chairperson of Computer Engineering & Science Department

Case Western Reserve University seeks to fill the position of Chairperson of the Department of Computer Engineering and Science. Scholarly and/or professional qualifications for appointment to the tenured faculty is required. Applicants with a record of scholarly research, an appreciation of the balance between teaching and research and the ability to develop research sponsorship are invited to apply.

CWRU is a private research university with a total enrollment of about 8000 students. The campus is located in University Circle which is a corporation of over 50 non-profit institutions, including nationally known museums and the home of the Cleveland Orchestra. The Department has 11 faculty members, 167 undergraduate majors, and 107 full-time graduate students. Of those graduate students, 44 are in the Ph.D. program. Faculty research efforts are concentrated in databases, programming methodology, computer aided design of VLSI systems, graphics, logic programming and intelligent systems. The education and research programs of the department are supported by good computing facilities including a Data General MV/10000, a VAX-11/780 and a network of workstations (mostly Suns). The department facilities are connected to all other major campus computing facilities by a high speed local area network which also provides access to national networks.

Applications will be accepted until June 15, 1988. Please send letter of application and resume to:

Dr. Robert E. Collin
Dean of Engineering
Case Western Reserve University
719 Crawford Hall
Cleveland, Ohio 44106

In employment, as in education, Case Western Reserve University is committed to equal employment and affirmative action.

BOOK REVIEWS

Editor: Wiley McKinzie, School of Computer Science and Technology, Rochester Institute of Technology, Rochester, NY 14623; Compmail, w.mckinzie; CSnet, wrm@rit

OSI

Handbook of Computer Communications Standards: Volume 1

William Stallings (Macmillan Publishing Co., New York, 1987, 322 pp., $34.95)

William Stallings is a well-known writer in the area of computer communications and networking. This book is the first volume of a series designed to present current computer communications standards in a comprehensive and understandable manner. The *Handbook of Computer Communications Standards: Volume 1* is a tutorial presentation of the OSI model, including protocol implementations of the various layers. The book is divided into ten chapters, including a general overview of protocol and communications standards, a description of the OSI model, and a chapter devoted to each of its layers. (The network layer is divided into the subnetwork interface and the internetwork interface, with each getting a chapter of its own.) Several of the chapters have appendixes covering important, but peripheral, topics. General appendixes at the end of the book describe the services and functions of the OSI layers and list the various standards cited in the book.

The first two chapters give the reader an overview of later topics and introduce the terminology and concepts that will be needed. They present a well written introduction to communications architectures and protocols, including the purpose for layering within protocols, and then explain and justify the seven layers of the OSI model. The second chapter concludes with a brief description of each of the OSI layers.

The rest of the book is devoted to detailed descriptions of the layers, including lists of each layer's OSI functions. In the first of these chapters, Stallings covers the physical layer and discusses the common physical layer standards: RS-232-C, RS-449/422-A/423-A and X.21. He explains how each of these standards was to be an improvement over its predecessor and introduces the common paradox in standards implementation: How do you implement standards and, at the same time, take advantage of advances in technology? The chapter concludes with a survey of data transmission issues such as the encoding of binary data and asynchronous and synchronous transmission.

The next chapter examines the data link layer and covers line configuration, flow control, and error control to examine a specific data-link protocol, HDLC. After covering HDLC, which is basically a point-to-point protocol, Stallings introduces three other protocols that address the multipoint issues: MLP (multilink procedure), an ISO protocol designed to operate over multiple lines; LLC (logical link control), a part of the IEEE 802 local area network protocols for multipoint links; and LAP-D, a part of the ISDN implementation for the D-channel of an ISDN link.

The next chapters cover the two parts of the network layer: the subnetwork interface and internetwork communications. The description of the subnetwork interface begins with an overview of the network layer and then discusses the interface between the DTE, which uses the communications services, and the DCE, which provides these services. This first chapter on the network layer concludes with a discussion of the X.25 packet network in relation to these network interface issues.

The next chapter describes the upper part of the network and covers internetwork communications. It begins with a discussion of the basics of internetworking, leading up to a comparison between X.75, the ISO standard, and IP (Internet Protocol), the DOD standard. Stallings then gives a more comprehensive description of X.75 and goes into connectionless internetworking. Next, he covers the ISO Internetwork Protocol before finishing up with a look at the internal organization of the network layer. The chapter contains appendixes on network layer addressing and the ISO checksum algorithm.

Stallings breaks his discussion of the transport layer into two pieces: connection-oriented operation and connectionless operation. Both the network layer and the transport layer can operate in a connection-oriented or connectionless mode, so Stallings concludes the chapter with a discussion of how each of the four possible cross products can be implemented.

The final three chapters of the book deal with the upper layers of the OSI model: the session, presentation, and application layers. There are few real implementations because these standards are relatively new, so the discussion mainly covers the services and functions of the layers. Stallings first describes the various services the session layer provides to the presentation and application layers. He then goes on to the presentation layer and discusses the ISO presentation service and its protocol. The chapter concludes with a discussion of the need for an abstract syntax and encoding rules to implement presentation layer services. The last chapter covers the application layer, including association control and commitment, concurrency, and recovery.

The book has two appendixes. The first is a concise list of the services and functions of each of the layers in the OSI model, and the second is a list of all the standards referenced in the book.

There is an enormous amount of detail in this book, as one would expect in a book dedicated to describing protocols. The presentation is well organized, with a consistent approach in each chapter. The reader interested in just a summary of each layer can read the introductory descriptions without delving into the protocol details. The descriptions are varied, including tables, state transition diagrams, and PDU formats, depending on the topic being covered, but are clearly presented.

I have tried to convey the breadth of Stallings' coverage; it would be much harder to convey its depth without substantial citations from the book. As a description of the OSI model, the book is excellent and I recommend it to anyone who needs to know about the model or any of its constituent layers. I look forward to the remaining two volumes in this series and hope they are up to the same standard of quality.

Lorne H. Schachter
Bell Communications Research

Programming Parallel Processors

Robert G. Babb II, ed., (Addison-Wesley Publishing Co., Reading, Mass., 1988, 380 pp., $37.75)

If juggling three balls is of difficulty 3 on a scale of 1 to 10, then juggling four balls is a difficulty of 20.—from a jugglers' handbook

Writing parallel software is a daunting task. It would be less daunting if you could learn your basic skills with a small number of parallel processes and then deal with the real complexity as the number of processes increased. But the knee of the difficulty curve for parallel software (unlike that for juggling) occurs immediately. You're in trouble as soon as there is any parallelism. You can expect to learn about deadlock, livelock, race conditions, and nondeterminism during your very first morning at work.

You need not be a software masochist, however, to be drawn toward parallel methods. The payback in compute power per dollar of cost is already attractive and likely to become more so. The cost of turning 64-bit Mflops on a monolithic architecture is something more than $10,000 per Mflop. Alternatively, some reasonably straightforward parallel architectures operate at less than $1,000 per Mflop, and some pipeline arithmetic units achieve still better price performance. Moreover, as the scale of parallel operation increases to the point that sets of processors can be integrated onto a single chip, the cost could drop drastically.

Until recently, parallel computing seemed germane only as a thesis topic for students who would subsequently turn to a life of Cobol. Now, a large and growing community of developers is actively exploring parallel approaches. If you are considering joining this community, you're going to need some help. Addison-Wesley's excellent *Programming Parallel Processors* is a good place to start.

The book was written (in parallel, of course) by nineteen authors under the direction of lead author and editor Robert Babb. After two introductory chapters on parallel processing, the work divides into eight units, each one oriented toward a different commercially available parallel machine: Alliant FX/8, BBN Butterfly, Cray X-MP, FPS T-Series, IBM 3090, Intel iPSC, Loral LDF 100, and Sequent Balance Series. Separate chapters describe the use of each machine on a typical parallel programming effort. Extensive appendixes for each machine give technical detail and excerpts from the manufacturers' parallel programming documentation. For many readers, the appendixes alone will be worth the price of the book.

The machine-specific chapters provide a clear picture of a first experience with each of the machines. Each of these chapters has its own "Pitfalls and Problems" section, for example, and each tells the story of a small project, including the results of the project and the project members' conclusions.

At a different level, *Programming Parallel Processors* tells the story of a rather elegant bit of management pulled off by editor Babb. He established a common context for his authors by designing a single application project for all to undertake and a single format for all the machine-specific chapters. He managed, as well, to impose an agreeable and surprisingly uniform writing style. Finally, he put it all together with a charming opening chapter. The result is an informative book, a compendium of details about the principal parallel machines, a comparative study on the use of these machines, and a reader's delectation ... all in parallel.

Tom DeMarco
The Atlantic Systems Guild

Software Metrics: Establishing a Company-Wide Program

Robert B. Grady and Deborah L. Caswell (Prentice-Hall, Englewood Cliffs, N.J., 1987, 288 pp., $35)

To paraphrase Charles Dickens, it is the best of books, it is the worst of books, it is full of insight, it is lacking in depth. In short, this is a book like many others on software metrics.

Software Metrics: Establishing a Company-Wide Program is the best historical account of the establishment of a Software Metrics Council at the Hewlett-Packard company. There are three threads in this history, and the skill with which they are interwoven makes it delightful reading. The threads are: (1) how to sell software metrics to top management, program managers and software engineers; (2) how the Software Metrics Council and its task forces operate; and (3) how to minimize software defects and increase software productivity through measurement. A clear presentation of the first two threads is essential for this history's primary objective: to demonstrate how to establish a company-wide metrics program. This objective is much too easily satisfied, though; the establishment of a standards committee (company-wide or otherwise) is neither a historic achievement nor a topic worth reading about. However, the third thread is of interest to software engineers and program managers.

Overall, *Software Metrics* lacks depth. The style is similar to the evening news: tidbits of information with many pictures.

The book is occasionally insightful. For example, the authors admit that current off-the-shelf models for software cost and schedule estimation may not be sufficient, that the Software Metrics Council may be collecting inappropriate metrics, and that it may take seven more years to determine if collecting metrics is worth the expense. In addition to this self-assessment, the authors describe an approach to their metrics problem that is well conceived, evolutionary and practical. First, the council selects a collectible set of metrics (size, people/time/cost, defects, difficulty, and communications) and gives project managers simple tools such as paper forms for collecting information and line counters for determining code size. The council then validates the returned data, adds it to the company-wide database, and adjusts the estimation models according to information in the database. As the metric program matures, the council can develop new tools, add and delete metrics, and, most importantly, adjust the model. Some early successes have been achieved by the Software Metrics Council, and both the authors and I think more will follow because of its approach.

Overall, though, *Software Metrics: Establishing a Company-Wide Program* lacks depth. The style is similar to the evening news: tidbits of information with many pictures. The following three examples will prove my point; there are many others.

First, the decision to use non-commented source statements (NCSS) over other line counting methods is adequately covered, but the real controversy is whether lines of code or some other metric (such as function points) should be used for software estimation. This controversy is barely revealed. A minimal explanation for using NCSS over function points is buried in a referenced paper's abstract in Appendix C.

Second, a discussion of the statistical validity and significance of collected data occupies only 4½ pages, one of which is a chart. This is not enough. The collection and validation of data is critical in quantifying defect rates, measuring productivity, and estimating future projects. Proper techniques in surveying and collecting data such as sampling, biases in the sample population, variance, and correlation analysis are barely mentioned. The reader is not given enough information to realize the great potential for getting skewed results from improperly collected data.

Third, the establishment of the Software Metrics Council is much too rosy to be believable. Unless the project was driven by a very high-level manager, the political and technical differences that normally accompany the formation of such a council would have been much more contentious.

I cannot recommend this book because the material lacks depth. However, it does have a few good points. The book thoroughly covers defect management and defect removal, and provides excellent references and abstracts. The authors include some good charts, forms, and slides. They also have organized the material well, with the exception of Chapters 16, 17, and 18, which do not follow the theme of the preceding chapters and therefore seem out of place. *Software Metrics: Establishing a Company-Wide Program* may be useful as a supplement to more detailed texts on software metrics.

Jeff Cartwright
The BMD Corporation

Software Engineering with Systems Analysis and Design

Donald W. Steward (Brooks/Cole Publishing Co., Monterey, Calif., 1987, 414 pp., $44.75)

This book provides both students and practitioners with an excellent integrated approach to software engineering and systems analysis and design. I agree with the author's statement that the book is "appropriate for classes in software engineering or systems analysis, as well as in the spectrum of computer science and management information classes." As a software developer, I found the topics discussed in the text helpful and applicable to my work.

Software Engineering with Systems Analysis and Design is organized into five parts. Part 1 introduces the principles used throughout the book; parts 2, 3, and 4 examine topics such as project management; and part 5 integrates all of the material into the standard phases of a software development project. I was impressed by the "tree" concept's role in supporting the author's integrated approach to software engineering and systems analysis and design. A tree structure can represent the broad requirements for a system as well as the details of a specific computer program. In other words, the end user of a system and the technician building parts of that system can understand different versions of the same tree.

The author gives the software engineering practitioner ways to use both conventional software development tools and some newer techniques. He discusses the familiar tools such as data-flow diagrams, structure charts, Warnier-Orr diagrams, structured English, and pseudocode. The methods are used at different times and in different ways in software development projects, and the author addresses the difficulties these differences present. I found his approach quite refreshing. The author suggests there are many valid ways of looking at the software development process, and we need to integrate these methods into a usable framework.

Part 1 of the book introduces software engineering, including such topics as "Why Do We Need Software Engineering?" and "Managing with the Project Life Cycle." There is a section on the importance of *expectations* in the software development process. As a practitioner, I continually remind myself of the importance of expectations and the need for precision when I document them.

There are many valid ways of looking at the software development process and we need to integrate these methods into a usable framework.

Part 2 reviews the classic methods for representing systems, such as data flow diagrams. This section also has an excellent introduction to tree structures. I was not familiar with this form of tree structures, but I found the author's explanation easy to understand and I can see the power and applicability of these trees. This topic alone makes the book worth reading.

Part 3 is an overview of software development project management, including these important topics: "Managing People and Expectations," "Estimating the Project," and "Scheduling and Controlling the Project." I appreciate any assistance I can get in project estimation.

Part 4 briefly surveys some of the technologies needed by software engineers to solve software problems (developed in more detail in other sources). The topics include "Principles for Developing User-Friendly Interfaces" and "Rapid Prototyping."

Part 5 examines each phase of the software development process in detail and applies the information introduced in the first four parts.

This is a very good book for students and practitioners. For the student in all of us, a case study in an appendix gives us an opportunity to apply the new techniques learned in the main sections. I highly recommend this book for software engineers, aspiring software engineers, and all software development professionals who do not yet realize they are software engineers.

Ronald L. Sigrist
Applied Management Systems

The Elements of Artificial Intelligence: An Introduction Using Lisp

Steven L. Tanimoto (Computer Science Press, Rockville, Md., 1987, 527 pp., $35.95)

This book provides an introduction to the principles and techniques of artificial intelligence. It is suitable as an undergraduate or first-year graduate text. There is ample material for a two-semester course, or a one-semester course consisting of selected topics. The book also could be used as a self-study course because it is reasonably self-contained. The average computer professional would have little difficulty with the prerequisite knowledge, which includes high-school math, some programming and computer experience, and some exposure to mathematical logic (at least propositional calculus and, preferably, predicate calculus). A knowledge of data structures also would help. The author integrates Lisp examples into most of the book's exposition, and the key algorithms and principles are illustrated by Lisp programs. The author believes in learning by doing, hence the programming emphasis throughout the text and in the exercises.

The book's introductory chapter is followed by chapters on programming in Lisp, productions and matching, knowledge representation, search, logical reasoning, and probabilistic reasoning. Chapters on learning, natural-language understanding, vision, expert systems, and a look into the future complete the book.

A question I asked myself in reviewing this book was: Considering the excellent texts on artificial intelligence on the market (such as P.H. Winston, *Artificial Intelligence*, Second Edition, Addison-Wesley, 1984; E. Charniak and D. McDermott, *Introduction to Artificial Intelligence*, Addison-Wesley, 1985; and E. Rich, *Artificial Intelligence*, McGraw-Hill, 1983), what does this book add to the literature? One consideration is that artificial intelligence is a fast-moving field and a new, up-to-date book is always welcome.

The integration of programming into the text is aided by additional materials. (A Lisp interpreter for IBM PCs and compatibles and the source for all the Lisp programs in the text can be purchased from SoftWave.)

Other outstanding features of the book are extensive exercises and an excellent bibliography at the end of each chapter. In addition, a section summarizing the literature precedes each bibliography, which should be helpful for further study.

The book also offers fine chapters on reasoning and vision that go to an unusual depth for an introductory text (for example, there is a discussion of Herbrand's Theorem as part of a discussion of the theoretical underpinnings of resolution principles). The vision chapter is quite extensive and enjoyable, perhaps because the author's principle area of research is computer vision. He is also editor-in-chief of *IEEE Transactions on Pattern Analysis and Machine Intelligence* and is on the editorial board of *Pattern Recognition*.

On the debit side, this book probably won't suit those people who want to learn about artificial intelligence without the benefit of Lisp, despite the quantity of interesting material without Lisp examples. The brevity of the chapter on expert systems also took me by surprise, particularly in view of the length of the chapters preceding it. Overall, however, the book is well written and a valuable addition to the field.

Joseph M. Kusmiss
IBM, Academic Information Systems

NEW LITERATURE

C programming book revised. The second edition of *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie will be published this month by Prentice Hall (cloth, ISBN 0-130110370-9, 284 pp., $40; paper, ISBN 0-13-110362-8, $28). The new edition will be based on the draft-proposed ANSI C Standard, and will include a C reference manual, an appendix describing the standard library, and an appendix summarizing changes between the first edition and the proposed standard. The first edition will remain in print for those programmers working with C compilers developed prior to the ANSI C draft proposal.

For more information, contact the Simon and Schuster Higher Education Group, Prentice Hall Building, Englewood Cliffs, NJ 07632; (201) 592-2000.

Robotics directory updated. The second *Robotics Today Product and Services Directory* offers a list of robot manufacturers and systems houses grouped by types of robotic system applications; a list of companies grouped according to the type of products and/or services offered; and a master alphabetical list of more than 1,000 companies. A separate compilation lists major organizations involved with the development and implementation of robotics-related technology. The publication is available for $24.95 through Colleen Nault, Society of Manufacturing Engineers, One SME Drive, P.O. Box 930, Dearborn, MI 48121; (313) 271-1500, extension 475.

Compact Disc Interactive: A Designer's Overview by the technical staff of Phillips International has been published by the McGraw-Hill Book Co. The 256-page book provides specific guidelines for designing CD-I applications; preparing audio, video and text/data material for CD-I mastering and replication; and working with the CD Real Time Operating System and the Invasion multiuser interface. The book is being published simultaneously by Kluwer Technical Books for distribution in Western Europe. Contact the McGraw-Hill Book Co., Professional and Reference Division, Engineering and Science Group, 11 W. 19th St., New York, NY 10011.

Micros Plus: Educational peripherals (ISBN 0-444-70380-2, 220 pp., $58) is the proceedings of the IFIP TC3 WG3.3 Working Conference on the Educational Implications of Connecting Tools and Devices to Microcomputers. Papers describe projects involving videodisc, Lego construction kits, concept keyboards, modems, speech input/output, and music synthesis. The proceedings examine impact on students, teachers, and researchers in elementary, secondary, and tertiary education. Contact Elsevier Science Publishing Co., P.O. Box 1663, Grand Central Station, New York, NY 10163.

ADVERTISER INDEX

Academic Press, Inc.	9
ACM Siggraph '88	7
CACI	1
CASE '88	Cover IV
Compcon '89	10
Computer Architecture Symposium	94-98
Computer Society Membership	115
Computer Society Ombudsman	135
Contel Technology Center	16
CVPR '88	81
Data Engineering Conference	70
Distributed Computing Systems Conference	124-125
GTE Laboratories	Cover II
Index Technology Corporation	68-69
Institute for Defense Analyses	93
Interactive Development Environments	57
Linotype Company	11
Meta Software Corporation	121
Multi-Valued Logic Technical Committee	134
National Business Employment Weekly	80
Parallel Processing Symposium	131
Software Engineering Conference	82
Sydetech, Inc.	55
Visible Systems Corporation	12
Classified Advertising	137-139

FOR DISPLAY ADVERTISING INFORMATION CONTACT

Northern California and Pacific Northwest: Don Farris Co., 161 W. 25th Ave., Ste. 102B, San Mateo, CA 94403; (415) 349-2222.
Jack Vance, P.O. Box 3205, Saratoga, CA 95070; (408) 741-0354.
Southern California and Mountain States: Richard C. Faust Co., 24050 Madison St., Ste. 100, Torrance, CA 90505; (213) 373-9604.
Southwest: The House Co., 3000 Weslayan, Ste. 345, Houston, TX 77027; (713) 622-2868.
Midwest: Robert Acker & Assoc., 480 Central Ave., Northfield, IL 60093; (312)441-6050.
East Coast: Hart Associates, PO Box 339, 42 Lake Blvd., Matawan, NJ 07747; (201) 583-8500.
New England: Arpin Associates, PO Box 227, Weston, MA 02193; (617) 899-5613.
Southeast/Telemarketing: Kay Young & Assoc., 2081 Business Center Dr., Suite 180, Irvine, CA 92714; (714) 551-4924.
Europe: Heinz J. Gorgens, Parkstrasse 8a, D-4054 Nettetal 1 - Hinsbeck (F.R.G); (0 21 53) 8 99 88.

Advertising Director: Dawn Peck.
For production information, conference, and classified advertising, contact Heidi Rex or Marian Tibayan.
For reader service inquiries, contact Heidi Rex.

Computer, 10662 Los Vaqueros Cir., Los Alamitos, CA 90720; (714) 821-8380.

PRODUCT INDEX

	RS #	Page#
BOARDS		
Display adapter card	25	113
Image processor	34	117
CIRCUITS		
Coprocessor platform	40	118
Graphics card	45	118
IC announcements	120-132	122
Seven-chip set	31	116
COMPONENTS		
Microsystems announcements	135-147	123
CONFERENCES & COURSES		
ACM Siggraph '88	—	7
CASE '88	—	Cover IV
Compcon '89	—	10
Computer Architecture	—	94-98
CVPR '88	—	81
Data Engineering	—	70
Distributed Computing Systems	—	124-125
Parallel Processing	—	131
Software Engineering	—	82
DATA COMMUNICATIONS		
Network protocol	41	118
I/O RELATED EQUIPMENT		
Laser printer	50	119
PUBLISHERS		
Career Opportunities	8	80
CS membership	—	115
CS ombudsman	—	135
Journal	1	9
Screen editor	3	55
Technical committee information	—	134
RECRUITMENT		
Classified advertising	—	137-139
Technical professional	—	Cover II, 1, 11, 16, 93
SOFTWARE		
C compiler	43, 54	118, 120
Evaluation program	24	113
Fortran compiler	51, 55	120
Mac software	46-49	119
Pascal compiler	53	120
SYSTEMS		
Computer	20, 21, 36, 37, 52	109, 111, 117, 120
Development tool	5, 38, 39, 44, 56	68-69, 118, 120
Graphics supercomputer	30	116
Graphics/text processor	2, 7	12, 121
Graphics workstation	33	117
Operating environment	22-23	112
Parallel processing system	35	117
Software design tool	42	118
Software environment	32	116
Tool environment	4	57

READER SERVICE CARD

COMPUTER

April 1988 issue

(Card void after September 1988)

Name____________________

Title____________________

Company____________________

Address____________________

City__________ State ___ ZIP______

Country________ Phone (___) ______

Please send

(Circle those you want):

- **201** Publications catalog
- **202** Membership information
- **203** Student membership information
- **204** Application for senior membership

Reader interest

(Circle what you liked, add comments on the back)

- **A** Guest Editor's Introduction
- **B** MVL Tutorial
- **C** Comparison of ICs
- **D** Multiplier Chip
- **E** CCD Circuits
- **F** MVL and PLAs
- **G** Fuzzy Logic

- **H** President's Message
- **J** Letters to the Editor
- **K** Standards
- **L** Computer Society News
- **M** Update
- **N** New Product Reviews
- **P** New Products
- **Q** IC/Microsystem Announcements
- **R** Conferences
- **S** Calendar/Call for Papers
- **T** Book Reviews
- **U** New Literature

Product inquiries

1

(circle the numbers for products and advertisers you want more information on)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	191
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

READER SERVICE CARD

COMPUTER

April 1988 issue

(Card void after September 1988)

Name____________________

Title____________________

Company____________________

Address____________________

City__________ State ___ ZIP______

Country________ Phone (___) ______

Please send

(Circle those you want):

- **201** Publications catalog
- **202** Membership information
- **203** Student membership information
- **204** Application for senior membership

Reader interest

(Circle what you liked, add comments on the back)

- **A** Guest Editor's Introduction
- **B** MVL Tutorial
- **C** Comparison of ICs
- **D** Multiplier Chip
- **E** CCD Circuits
- **F** MVL and PLAs
- **G** Fuzzy Logic

- **H** President's Message
- **J** Letters to the Editor
- **K** Standards
- **L** Computer Society News
- **M** Update
- **N** New Product Reviews
- **P** New Products
- **Q** IC/Microsystem Announcements
- **R** Conferences
- **S** Calendar/Call for Papers
- **T** Book Reviews
- **U** New Literature

Product inquiries

2

(circle the numbers for products and advertisers you want more information on)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	191
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

READER SERVICE CARD

COMPUTER

April 1988 issue

(Card void after September 1988)

Name____________________

Title____________________

Company____________________

Address____________________

City__________ State ___ ZIP______

Country________ Phone (___) ______

Please send

(Circle those you want):

- **201** Publications catalog
- **202** Membership information
- **203** Student membership information
- **204** Application for senior membership

Reader interest

(Circle what you liked, add comments on the back)

- **A** Guest Editor's Introduction
- **B** MVL Tutorial
- **C** Comparison of ICs
- **D** Multiplier Chip
- **E** CCD Circuits
- **F** MVL and PLAs
- **G** Fuzzy Logic

- **H** President's Message
- **J** Letters to the Editor
- **K** Standards
- **L** Computer Society News
- **M** Update
- **N** New Product Reviews
- **P** New Products
- **Q** IC/Microsystem Announcements
- **R** Conferences
- **S** Calendar/Call for Papers
- **T** Book Reviews
- **U** New Literature

Product inquiries

3

(circle the numbers for products and advertisers you want more information on)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	191
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

Editorial comments

I liked:

I disliked:

I would like to see:

For reader service inquiries, use other side

PO box address for
reader service cards only

PLACE
STAMP
HERE

Computer
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91615-6508
USA

Editorial comments

I liked:

I disliked:

I would like to see:

For reader service inquiries, use other side

PO box address for
reader service cards only

PLACE
STAMP
HERE

Computer
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91615-6508
USA

Editorial comments

I liked:

I disliked:

I would like to see:

For reader service inquiries, use other side

PO box address for
reader service cards only

PLACE
STAMP
HERE

Computer
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91615-6508
USA

THE COMPUTER SOCIETY

A member society of the Institute of Electrical and Electronics Engineers, Inc.

Executive Committee

President: Edward A. Parrish, Jr.*
Vanderbilt University
School of Engineering
Nashville, TN 37235
(615) 322-2762

President-Elect: Kenneth R. Anderson*

Vice Presidents
Standards: Helen M. Wood (1st VP)*
Conferences and Tutorials: Joseph E. Urban (2nd VP)*
Area Activities: Willis King†
Education: Gerald L. Engel†
Membership and Information: Merlin G. Smith†
Publications: James H. Aylor†
Technical Activities: Laurel V. Kaleda

Secretary: Duncan H. Lawrie
Treasurer: Barry W. Johnson†
Past President: Roy L. Russo*
Division V Director: Harriett Rigas†
Division VIII Director: H. Troy Nagle, Jr.†
Executive Director: T. Michael Elliott†

*voting member of the Board of Governors
†nonvoting member of the Board of Governors

Board of Governors

Term Ending 1988	Term Ending 1989
Mario Barbacci	Bill D. Carroll
Victor Basili	Lansing Hatfield
Paul Borrill	Duncan H. Lawrie
Lorraine M. Duvall	David Pessel
Michael Evangelist	Susan L. Rosenbaum
Allen L. Hankinson	Sallie V. Sheppard
Laurel V. Kaleda	Bruce D. Shriver
Ted Lewis	Harold S. Stone
Ming T. Liu	Akihiko Yamada
Earl E. Swartzlander, Jr.	Marshall C. Yovits

Next Board Meeting

June 17, 1988—8:30 a.m.
Anaheim Sheraton, Anaheim, CA

Senior Staff

Executive Director: T. Michael Elliott
Editor and Publisher: True Seaborn
Director, Computer Society Press: Chip G. Stockton
Director, Conferences: Anne Marie Kelly
Director, Finance and Administration: Mary Ellen Curto

Computer Society Offices

Headquarters Office
1730 Massachusetts Ave. NW
Washington, DC 20036-1903
Phone: (202) 371-0101
Telex: 7108250437 IEEE COMPSO

Publications Office
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
Membership and General Information: (714) 821-8380
Publications Orders: (800) 272-6657

European Office
13, Avenue de l'Aquilon
B-1200 Brussels, Belgium
Phone: 32 (2) 770-21-98
Telex: 25387 AVVALB

Use the Reader Service Card to obtain the following information:

- Membership application—student #203, others #202
- Periodicals subscription form for individuals #200
- Periodicals subscription form for organizations #199
- Publications catalog #201
- Standards working groups list #195
- COMPMAIL+ international electronic mail/database brochure #194
- Technical committee list/application #197
- Chapters lists, start-up procedures—student/regular #193
- Student scholarship information #192
- Awards description/nomination forms #198
- Volunteer leaders/staff directory #196
- IEEE senior member application #204

Purpose

The Computer Society advances the theory and practice of computer science and engineering, promotes the exchange of technical information among 90,000 members worldwide, and provides a wide range of services to members and nonmembers.

Membership

Members receive the acclaimed monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others seriously interested in the computer field.

Publications and Activities

Computer. An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field; plus news, conferences, calendar, interviews, new products, etc.

Periodicals. The society publishes six magazines and three research transactions. Refer to membership application or request information as noted above.

Conference Proceedings, Tutorial Texts, Standards Documents. The Computer Society Press publishes more than 100 titles every year.

Standards Working Groups. Over 100 of these groups produce IEEE standards used throughout the industrial world.

Technical Committees. Over 30 TCs provide newsletters, interaction with peers in specialty areas, and have direct influence on standards, conferences, education, etc.

Conferences/Education. The society holds about 100 conferences each year and sponsors many educational activities, including computing sciences accreditation.

Chapters. Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

European Office

Payments for Computer Society membership and publication orders are accepted by cheques in Belgian, British, German, Swiss, or U.S. currency, or by American Express, Eurocard, MasterCard, or Visa credit cards.

Ombudsman

Members experiencing problems—magazine delivery, membership status, unresolved complaints—may write to the ombudsman at the Publications Office.

Also see membership application in this magazine.

CASE '88

Second International Workshop on Computer-Aided Software Engineering

Hyatt Regency Cambridge
Cambridge, Massachusetts
July 12–15, 1988

In Cooperation With:

THE COMPUTER SOCIETY OF THE IEEE

Call For Papers

The field of Computer-Aided Software Engineering (CASE) has recently emerged as a commercially-viable widespread application of software engineering techniques and computer technology to information systems development.

CASE '88 provides a forum for solid, detailed exchange of ideas among key practitioners, researchers, developers, and leading-edge users in the field, resulting in meaningful goals for the advancement of CASE technology in the next five years.

Working groups will review the current state-of-the-art, discuss present research directions, and consider future requirements in the following topic areas:

- *Management Issues*
  - *—Management of CASE*
  - *—Technology Transfer*
  - *—Support for Top-level Management & Strategic Planning*
- *Methods for Information Systems Development*
  - *—Integration of Methods*
  - *—Support between Methods & Tools*
  - *—Real-time Development Methods*
  - *—Object-Oriented Design*
- *Enabling Technologies*
  - *—A.I. / Expert Systems*
  - *—Programming Language Evolution*
  - *—Database / Knowledge Base Architectures*
  - *—Ada™ Technology*
  - *—User Working Environment / CHI*
- *Reverse Engineering*
- *Standards for CASE*

General Chair

Elliot Chikofsky
Index Technology Corporation

Program Committee

Ronald J. Norman
San Diego State University

Hasan Sayani
Advanced Systems Technology Corporation

Jerrold Grochow
American Management Systems

John O. Jenkins
Imperial College

Burt Rubenstein
Index Technology Corporation

Tony Wasserman
Interactive Development Environments

Thomas P. Cullinane
Northeastern University

Karl Lieberherr
Northeastern University

Bob Carasik
Pacific Bell

Peter Mager
PSM Associates

Lonnie Bentley
Purdue University

Priscilla Fowler
Software Engineering Institute

William Cureton
Sun Microsystems

Kate Ehrlich
Symbolics

Alan R. Hevner
University of Maryland

W. Richards Adrion
University of Massachusetts

Lt. Michael F. Merriman
U.S. Air Force ESD

Thomas Browdy
Washington University

Attendance is limited. Prospective attendees should submit 5 copies of a position paper (1 to 4 pages), discussion forum proposal, or extended abstract (up to 10 pages) on future directions of CASE technology related to one or more of the topic areas, ***by April 11, 1988.***

Machine-readable submissions are preferred.

Program Coordinator

Pamela Meyer
CASE '88
c/o Index Technology Corporation
One Main Street
Cambridge, MA 02142

(617) 494–8200 ext. 1988

CSNET: Chikofsky@VAXE.COE.Northeastern.EDU

European Coordinator

John O. Jenkins
Imperial College
School of Management
London SW7 2PG, United Kingdom
01–589–5111 ext. 7112

Exhibits Chair

Victor Barlow
Purdue University
Dept. of Computer Technology
Knoy Hall, Room 242
West Lafayette, IN 47907
(317) 494–4546

Sponsored by:
Index Technology Corporation

In Cooperation With:
Imperial College
Purdue University
Northeastern University
Washington University CSDP
Greater Boston Chapter ACM